

Backwards Compatibility Definition

There are ~~two~~three areas of Backwards Compatibility. These are defined below:

- Pure Backwards Compatibility – implies that interface specification has NOT been modified and therefore, no recompile is necessary. Also, no behavior on the NPAC SMS has been modified to provide any change to the previously existing functionality accessible over the interface.
- Interface and Functional Backwards Compatibility – implies that the interface may have been modified, however the changes are such that ~~only a recompile is necessary~~ no action is required for the Service Provider to remain backward compatible. ~~However, A~~any new functionality is optionally implemented, and would require a recompile and possibly functional software changes by the Service Provider ~~to~~ by-accessing the newly defined features over the interface. Also, no changes may be made to any existing interface functionality that will require modifications to SOA and/or LSMS platforms.
- Re-Compile Only Backwards Compatibility – implies that the interface has been modified, however the changes are such that only a recompile is necessary to remain backward compatible. Any new functionality is optionally implemented by accessing the newly defined features over the interface. Also, no changes may be made to any existing interface functionality that will require modifications to SOA and/or LSMS platforms.

The general guideline is that subsequent releases of a major release (e.g., 2.0, 2.1, 2.1.1, etc.) must support Pure Backward Compatibility. Also, major releases should support at least one version of Interface and Functional Backward Compatibility (i.e., R3.0 should be Interface and Functional Backward Compatible to R2.0). The objective is that all releases remain Interface and Functional Backward Compatible, if possible.