Adapt Framework Standards definition

	Core bundle	**Contrib**	**Community code**
What is it?	Default installation of the Adapt Framework if done via Grunt. The code is written and maintained by the core team. Includes the following features: • adapt_framework • adapt-cli • adapt-contrib-vanilla • adapt-contrib-text • adapt-contrib-narrative • adapt-contrib-media • adapt-contrib-hotgraphic • adapt-contrib-blank • adapt-contrib-accordion • adapt-contrib-graphic • adapt-contrib-matching • adapt-contrib-textInput • adapt-contrib-mcq • adapt-contrib-gmcq • adapt-contrib-assessment • adapt-contrib-trickle • adapt-contrib-tutor • adapt-contrib-boxMenu • adapt-contrib-spoor • adapt-contrib-pageLevelProgress • adapt-contrib-resources • adapt-contrib-slider	Plug-ins (components, extensions, themes and menus) written and maintained by the core team but not included in the default installation.	Plug-ins (components, extensions, themes and menus) or other code written and maintained by the community. **A general note:** The Adapt Learning team might choose to add popular plug-ins to be added to contrib or the core bundle. As part of this, the team would also invite the developer / maintainer into the core team as a maintainer of the specific plug-in they have built. For acceptance into contrib / the core bundle, plug-ins are required to adhere to the standards referenced in this document. We therefore highly recommend anyone to work as closely to our standards as possible in order to facilitate as simple a transition as possible.

	Core bundle	**Contrib**	**Community code**
Owned and maintained by	Core Adapt Learning team	Core Adapt Learning team	A community member (referred to in this document as developer / maintainer)
Required to be **cross-platform compatible?**	Yes Aiming for ~90% device coverage. This is defined using Statcounter with the following rules: • Platform: Desktop, Mobile, Tablet • Statistic: Browser Verison (combine Chrome & Firefox) • Region: Worldwide • Period: Last 12 months Each release of the Adapt Framework will have it's own cross platform compatibility spec. For version 1.1, the supported platforms are: • Chrome (all; 36.21%) • Firefox 5+ (14.69%) • Android (6.29%) • IE10 (6.28%) • IE8 (5.93%) • iPhone (4.29%) • IE9 (3.74%) • Safari iPad (3.59%) • IE11 (3.1%) • Opera (3.07%) • Nokia (1.23%) • Safari 6 (1.16%) • **TOTAL: 89.58%**	No The Adapt Learning core team will aim to make Contrib plug-ins as cross platform compatible as possible but there is no guarantee of this. Each plug-in should include the platforms the plug-in is expected to work on as part of its metadata.	At the discretion of the developer / maintainer We encourage people to supply this information as metadata (e.g. on the main readme page for the repository in Github)

	Core bundle	**Contrib**	**Community code**
Required to be **responsive / work on mobile devices?**	Yes The functionality is responsive and works on mobile devices.	Yes The functionality is responsive and works on mobile devices.	At the discretion of the developer / maintainer
Required to **adhere Adapt Learning coding standards?**	Yes The Adapt Learning coding standards are defined here.	Yes The Adapt Learning coding standards are defined here.	At the discretion of the developer / maintainer We strongly encourage developers / maintainers to write code in line with our coding standards.
Required to include sufficient **unit test coverage?**	Yes The code needs to have sufficient unit test coverage and pass automated Travis unit tests. The required standard is defined here: *At the time of writing this document, the approach on unit tests still has to be documented.*	Yes The code needs to have sufficient unit test coverage and pass automated Travis unit tests. The required standard is defined here: *At the time of writing this document, the approach on unit tests still has to be documented.*	At the discretion of the developer / maintainer We encourage developers / maintainers to include unit tests in line with our coding standards.
Required to include **documentation?**	Yes All features need to be well documented, in terms of technical as well as user documentation.	Yes All features need to be well documented, in terms of technical as well as user documentation.	At the discretion of the developer / maintainer We strongly encourage developers / maintainers to write good documentation for their work.
Required to meet **accessibility standards?**	Yes WAI AA accessibility as of version 1.2 of the Adapt Framework.	No The Adapt Learning core team will aim to make Contrib plug-ins WAI AA accessible but there is no guarantee of this.	At the discretion of the developer / maintainer We encourage developers / maintainers to think about accessibility issues. Relevant

	Core bundle	**Contrib**	**Community code**
		Each plug-in should include a statement on whether the plug-in is accessible (and if so, to what degree) as part of its metadata.	guidance can be found here. Choose settings: • Levels: A, AA
Required to meet **Performance tests?**	Needs to pass performance tests defined as: For a given standard course (to be defined) not served by an LMS, content must be downloaded and rendered (details to be defined) within the following timescales • 10 seconds on an iPhone running on an Edge (2G) network • 1 second on Chrome (or whatever the most common User Agent is) running on a broadband (2MBit/s - as defined by Government) connection	No The Adapt Learning core team endeavor to test and update plug-ins as soon as possible after a new release has been made.	At the discretion of the developer / maintainer
Required to support **Right-to-left (RTL) layout?**	Currently in discussion with Kineo Israel on whether we can collaborate and add this to the roadmap.	No Each plug-in should include a statement on whether the plug-in supports RTL.	At the discretion of the developer / maintainer
What metadata should be provided?	• Version number • Author / maintainer • Which version of the core framework this works with • Cross-platform coverage • Accessibility support / standard • RTL support	• Version number • Author / maintainer • Which version of the core framework this works with • Cross-platform coverage • Accessibility support / standard • RTL support	• Version number • Author / maintainer • Which version of the core framework this works with • Cross-platform coverage • Accessibility support / standard • RTL support
Required to be	Yes: very thorough	Yes: through peer review	At the discretion of the developer /

	Core bundle	**Contrib**	**Community code**
thoroughly tested	Producing stable, high-quality software is one of the Adapt Learning core principles. We follow a peer review process, requiring three approvals (+1's) before a code change can be merged. We issue a pre-release for testing by the community and actively seek feedback. Each minor version (X.**Y**.Z, e.g. 1.1, 1.2 etc.) release goes through a thorough testing cycle according to the standards defined in this document.	Producing stable, high-quality software is one of the Adapt Learning core principles. We follow a peer review process, requiring three approvals (+1's) before a code change can be merged. We inform the community of new components and encourage the community to test these and report issues.	maintainer We strongly developers / maintainers to tests their work and also to use the community to help with this. The engage the community, developers / maintainers should consider announcing a new component on the community site forum and asking for feedback and help with testing.
How to make changes?	Gitflow process and do peer review. In summary, a pull request should be created and requires three core member approvals (+1s) before it will be merged. Permissions for core bundle repositories are tightly managed. The Gitflow process document exists but may be slightly out of date at the time of writing this document.	Gitflow process and do peer review. In summary, a pull request should be created and requires three core member approvals (+1s) before it will be merged. Permissions for core bundle repositories are tightly managed. The Gitflow process document exists but may be slightly out of date at the time of writing this document.	At the discretion of the developer / maintainer
How to report issues?	Via Github against the relevant repository For issues reported against standards set out in this document, the Adapt Learning core	Via Github against the relevant repository	At the discretion of the developer / maintainer We suggest using Github against the relevant

	Core bundle	**Contrib**	**Community code**
	team endeavor to prioritize to fix the issues reported.		repository
Registered with the Adapt registry?	Yes All core bundle plug-ins are (have to be) registered with the Adapt registry in order to enable developers to install them easily via the command line.	Yes All contrib plug-ins are registered with the Adapt registry in order to enable developers to install them easily via the command line.	At the discretion of the developer / maintainer We encourage developers / maintainers to register plug-ins with the Adapt registry in order to enable users to install these easily via the command line. The registry also serves as a vehicles which will (in time): • Check for and require metadata • Make the component available to the authoring tool (once developed)
Required to be **backwards compatible?**	Yes, as of version 1.1.0 Backwards compatibility is required for plug-ins as of the later of the two dates following: • v1.1.0 of the Adapt Framework or • date of incorporation of the plug-in into the core bundle	Yes, as of version 1.1.0	At the discretion of the developer / maintainer We encourage the developers / maintainers to list the core framework versions the plug-in works with.
Will it be updated when the core framework changes?	Yes All plug-ins in the core bundle will work with new releases at the time the release is made.	Yes The Adapt Learning core team endeavor to test and update plug-ins as soon as possible after a new release has been made.	At the discretion of the developer / maintainer We encourage developers / maintainers to test and update plug-ins after each new release. We expect and hope that required changes will be minimal.

	Core bundle	**Contrib**	**Community code**
Version numbering of component parts	We propose to mirror the version number of the core Adapt Framework in each plug-in up to and including the minor version (i.e. **X.Y.**<framework maintenance release>). This means for version 1.1.0 and all following maintenance releases of the core bundle, each plug-in will be versioned as **1.1.**<plug-in maintenance release>). Note: The last digit varies between the core framework. This way, it is easy to see by the version number whether a core bundle or contrib plug-in has been officially ported to the new core framework release. For core bundle plug-ins, the version number of the plug-ins will change at the time of a new major or minor version release of the Adapt Framework.	Same versioning approach as for core bundle plug-ins. The only difference is that contrib plug-ins may not be updated at the time of a new release. However, they may still work without modification. The increased version number is simply an indicator that the plug-in has been checked against the new version.	At the discretion of the developer / maintainer