Zowe v2 Conformance Test Evaluation Guide

The Zowe v2 Conformance Test Evaluation Guide is a set of self-certifying and self-service tests to help the development community integrate and extend specific technology into the Zowe framework.Â

This guide describes the requirements of the four available conformance programs. Items marked **(required)** are required for an application to be conformant. Items marked **(best practice)** are considered best practices for conformant applications.

These Zowe Conformance criteria are applicable to the lastest Zowe v2 LTS Release.

- Zowe v2 Conformance Test Evaluation Guide
  - Zowe API Mediation Layer
    - Application Service
    - REST API Documentation
    - REST API Naming and Addressing
    - Service Requests and Responses
    - Authentication and Authorization
    - Versioning and Support
    - WebSocket Services
    - Directory and File Ownership Permissions
    - Packaging
    - Support
  - Zowe CLI
    - Infrastructure
    - Installation
    - Naming
    - Profiles
    - Support
    - Documentation
  - Zowe Application Framework
    - Packaging
    - Packaging Format and Manifest
    - Web UIs All
    - Web UI IFrame
    - Web UI Non-IFrame
    - UI Design
    - Localization and Internationalization (l10n and l18n)
    - App Server
    - Documentation
    - Logging
    - Encoding
    - Storage
    - Directory and File Ownership Permissions
    - Lifecycling as a Zowe address space
    - Support
  - Zowe Explorer for Visual Studio Code
    - General Extension
    - Extension Accessing Profiles

Zowe API Mediation Layer

Application Service

Item	Version	Required	Best Practice	Conformant	Criteria
1	v2	x			The product or applications that extends Zowe API ML must provide at least one API service registered with the Zowe API ML Discovery Service
2		Mark (a) or (b)			A service must be registered using one of the following methods [please mark which one applies (a) or (b)]:
	v2		x		a. Dynamic Registration
	v2	x			b . Static Definition
3	v2	x			The service must provide a default service ID that is prefixed by the organization/provider name. Examples of compliant service IDs: zowemonitoring, bcmjclcheck, ibmims, rocketterasam Examples of non-compliant service IDs: jclcheck, myims, mydb2 Note: The service ID is part of the URL
4	v2	x			The service ID must be configurable externally after deployment
5	v2	x			The service ID must be written in lower case, contain no symbols, and is limited to 64 characters
6	v2	x			Services must provide API ID to provide grouping of APIs provided by multiple services. The API ID is for example used for Automated CLI Client Configuration. The API ID must be prefixed by the organization/provider name. Example of the API ID: zowe.apiml.apicatalog Note: API ID isn't part of the URL
7	v2	x			The published service URL must follow the gateway URL conventions
8		Versioned			For versioned APIs, service URLs must contain a service ID in the first place in the path, before the service version (all formats). Example formats: (Mark only one section - Versioned or Non-Versioned)
	v2	x			- {serviceId}/api/v1 reserved for REST APIs
	v2	x			- {serviceId}/ui/v1 reserved for UIs
	v2	x			- {serviceId}/ws/v1 reserved for WebSockets
		Non-Versioned			For non-versioned APIs or APIs versioned differently (e.g. z/OSMF), use the following formats:
	v2	x			- {serviceId}/api reserved for REST APIs
	v2	x			- {serviceId}/ui reserved for UIs
	v2	x			- {serviceId}/ws reserved for WebSockets
	v2	x			- {serviceId}/graphql reserved for GraphQL APIs
9		Mark (a), (b) or (c)			Registration of the service must be performed via one of the supported methods: (Mark which one applies _a_, _b_, _c_)
	v2	x			a. Adding the static API definition YAML file path to zowe.yaml
	v2	x			b. Copying the static API definition YAML file to the workspace api-definitions directory
	v2	x			c. Dynamic registration of an application

REST API Documentation

Item	Version	Required	Best Practice	Conformant	Criteria
10	v2	x			Documentation is Swagger/Open API 2.0/Open API 3.0 compliant
11	v2	x			Every public resource is documented with a description of each end-point
12	v2	x			Every method of each public REST endpoint is documented
13	v2	x			Every method of each public REST endpoint is demonstrated by example. The examples are part of the documentation.
14	v2	x			Every parameter (headers, query parameters, payload, cookies, status codes etc.) is documented with definitions of all possible values and their associated meanings
15	v2	x			Every HTTP error code must be documented. If the endpoint has additional and more granular error codes only provide the documentation reference.

REST API Naming and Addressing

Item	Version	Required	Best Practice	Conformant	Criteria
16	v2		x		Encoded slash is not used
17	v2	x			The service interprets values independent of their URL encoding
18	v2		x		lowerCamelCase is used for names of resources, parameters, and JSON properties

Service Requests and Responses

Item	Version	Required	Best Practice	Conformant	Criteria
19	v2		x		REST API - Request and response payloads are in JSON or binary data format
20	v2	x			REST API - - Since in JSON format, links are relative, links in responses payloads should not contain the schema, hostname, and port. Alternatively, an absolute link can be used, in which case the service must translate the link to the form that goes through the Gateway that is based on the X-Forwarded-* Headers.
21	v2	x			WebSocket (if applicable) - Service URIs contained in WebSocket messages payload are addressed through the API ML Gateway
22	v2	x			UI (if applicable) - The UI uses relative links and does not contain the schema, hostname, and port. Alternatively an absolute link can be used, in which case the service must translate the link to the form that goes through the Gateway that is based on the X-Forwarded-* Headers

Authentication and Authorization

Item	Version	Required	Best Practice	Conformant	Criteria
23	v2	x			Resources are protected by SAF authorization. All Single-Sign-On methods under item 23 are based on SAF authentication.
24	Please mark which apply (a), (b) or (c)				Services support Single-Sign-On using at least one of the following methods [please mark which apply (a), (b) or (c)]:
	v2	x			(a) Services accept PassTickets in the Authorization header of the HTTP requests using the basic authentication scheme (httpBasicPassTicket).
	v2	x			(b) Services accept Zowe JWT token in the cookie (zoweJwt)
	v2	x			(c) Services accept SAF Identity Token in the cookie (safIdt)

Versioning and Support

Item	Version	Required	Best Practice	Conformant	Criteria
25	v2		x		Service implementation follows the semantic versioning model

Item	Version	Required	Best Practice	Conformant	Criteria
26	v2	x			Last two major versions are supported by API services

WebSocket Services

Item	Version	Required	Best Practice	Conformant	Criteria
27	v2	x			WebSocket connection creation, all subsequent communication between WebSocket client, and server is routed through the API ML Gateway
28	v2	x			WebSocket connections are closed by the initiator through API ML Gateway

Directory and File Ownership Permissions

Item	Version	Required	Best Practice	Conformant	Criteria
29	v2	x			A conformant application must not modify the contents of the Zowe runtime USS directory and it must not change any directory or file permissions or ownership within the Zowe runtime
30	v2	x			A conformant application must not modify the permissions or ownership of a Zowe instance directory workspace.

Packaging

Item	Version	Required	Best Practice	Conformant	Criteria
Applicable if LIFECYCLED					Satisfy the following criteria to lifecycle a service with Zowe. By lifecycling we mean that Zowe Launcher will start, configure, validate and stop the service:
31	v2	x			If the extension is standalone (not bundled with other products), the extension convenience build (non-SMPE) MUST package itself into zip, tar, or pax format.
32	v2	x			The extension MUST package a manifest.yaml (or manifest.yml, or manifest.json) into final package.
33	v2	x			If the extension is standalone (not bundled with other products), the manifest file MUST be placed in the root directory after it's extracted or installed.
34	v2	x			The extension manifest MUST contain "name" definition.
35	v2		x		It's recommended to define "version" in the extension manifest.
36	v2		x		It's recommended to define "license" in the extension manifest .
37	v2		x		It's recommended to define "title" and "description" in the extension manifest.
38	v2		x		It's recommended to define "build" in the extension manifest to describe the source code information from version control systems.
39	v2		x		The extension is suggested to define "commands.install", "commands.configure", and/or "commands.start" to instruct Zowe how to install, configure or start in Zowe context.
40	v2		x		If the extension contains services that will be registered under API-ML, those services MUST be defined as entries of "apimlServices" in the extension manifest.

Support

Item	Version	Required	Best Practice	Conformant	Criteria
41	v2	x			Submitter describes how Support is provided and Support details are clearly documented

Zowe CLI

Infrastructure

Item	Version	Required	Best Practice	Conformant	Criteria
1	v2	x			Plug-in is constructed on the Imperative CLI Framework
2	v2	x			Plug-in is NOT run as a standalone CLI
3	v2	x			Plug-in commands write to stdout or stderr via Imperative Framework response.console APIs
4	v2	x			If plug-in requires gzip decompression support, leverage the Core CLI built-in support -- do NOT opt-out of the built-in gzip decompression support (specifically, the `mDecode` property of the Imperative RestClient must NOT be overridden).
5	v2	x			Plug-ins must not have an @zowe/cli peer dependency for improved npm@7 compatibility. The only peer dependencies that should be included are packages which are imported in the plug-in's source code (e.g., @zowe/imperative).
6	v2		x		In their Imperative config file (defined in the imperative.configurationModule property of package.json), plug-ins should make their imports as few and specific as possible. This can significantly decrease their load time (example).
7	v2		x		HTTP or HTTPS requests to REST APIs should use the @zowe/imperative RestClient instead of a direct dependency on a 3rd-party package like request or axios.

Installation

Item	Version	Required	Best Practice	Conformant	Criteria
8	v2	x			Plug-in is installable with the zowe plugins install command
9	v2	x			Plug-in is installable into the @zowe-vN-lts version of the core Zowe CLI and follows semantic versioning (where "N" = the latest "active" LTS release number)
10	v2	x			Plug-in is uninstallable via the zowe plugins uninstall command
11	v2	x			`@latest` should point to the same version as the most recent zowe lts tag (Note: for V2 it will be `@zowe-v2-lts`)

Naming

Item	Version	Required	Best Practice	Conformant	Criteria
12	v2	x			If the plug-in introduces a command group name, it does not conflict with existing conformant plug-in group names
13	v2		x		Names of CLI commands/groups/options must be in kebab case (lower case & hyphens). Names of properties in zowe.config.json should be camel case. Only alphanumeric characters should be used - `[a-zA-Z0-9-]+`.
14	v2	x			The following option names/aliases are reserved and must not be used: --response-format-json, --rfj, --help, -h, --help-examples, --help-web, --hw

Profiles

Item	Version	Required	Best Practice	Conformant	Criteria
15	v2	x			If the plug-in has unique connection details, it introduces a profile that lets users store these details for repeated use
16	v2		x		Plug-in users are able to override all profile settings via the command line and/or environment variables
17	v2	x			If the plug-in uses a Zowe API-ML integrated API, it (the plug-in) has an option named `base-path` in the profile to used to house the path of the associated service in the API ML.
18	v2	x			If the plug-in connects to a service, it must include the following profile properties AND they MUST be these exact properties (e.g. host, NOT hostname): 'host', 'port', 'user', 'password'

Item	Version	Required	Best Practice	Conformant	Criteria
19	v2	x			If the plug-in connects to a service, and the service supports logging in with a token, it must include the following profile properties AND they MUST be these exact properties: 'tokenType', 'tokenValue'
20	v2	x			If the plug-in connects to a service, and the service supports logging in with PEM certificates, it must include the following profile properties AND they MUST be these exact properties: 'certFile', 'certKeyFile'
21	v2	x			If the plug-in connects to a service, and the service supports logging in with PFX/P12 certificates, it must include the following profile properties AND they MUST be these exact properties: 'certFile', 'certFilePassphrase'
22	v2	x			If the plug-in provides an option to reject untrusted certificates, the property must be named "rejectUnauthorized". CLI option should be reject-unauthorized.
23	v2		x		The plug-in specifies options to be pre-filled by default in zowe.config.json once `zowe config init` has executed
24	v2	x			Plug-in supports team-profile config
25	v2	x			Plug-in supports base profiles
26	v2		x		When host, port, user, or password is missing for a particular command and no default value is set, the user is prompted for the argument. Host, user, and password should NOT have default values.
27	v2	x			To take advantage of the new 'zowe config auto-init' command, a plugin that works with a single-sign-on, APIML-compliant REST service MUST supply a new object within its plugin definition to identify that REST service. The new IImperative.apimlConnLookup object must be in the plugin's definition. That object includes the apiId and gatewayUrl of the corresponding REST service. The related REST service must also supply its apiId and gatewayUrl in the apiml section of its application.yml definition. Zowe-CLI automatically handles the apimlConnLookup object for the 'zosmf' service. Thus an apimlConnLookup object for 'zosmf' does not have to be specified within a plugin.

Support

Item	Version	Required	Best Practice	Conformant	Criteria
28	v2	x			Submitter describes how Support is provided and Support details are clearly documented

Documentation

Item	Version	Required	Best Practice	Conformant	Criteria
29	v2		x		Plug-in command help is contributed to this repo for the purpose of hosting the ecosystem web-help on Zowe Docs - https://github.com/zowe/zowe-cli-web-help-generator

Zowe Application Framework

Packaging

Item	Version	Required	Best Practice	Conformant	Criteria
1	v2	x			Every plugin must have a unique ID. The ID format follows java package naming conventions. The Zowe project reserves org.zowe
2	v2	x			Every plugin and each of its services must have a version
3	v2	x			Directory layout adheres to the App filesystem structure
4	v2		x		Source code is also recommended, but not required to adhere to the App filesystem structure for tooling consistency
5	v2	x			Plugins must be packaged as part of a Component, so that Component install & managment tooling can be used

Item	Version	Required	Best Practice	Conformant	Criteria
6	v2	x			Plugins must not be installed by calling install-app.sh directly, as Component tooling calls it instead. More information is available in "Packaging" conformance section

Packaging Format and Manifest

Item	Version	Required	Best Practice	Conformant	Criteria
7	v2	x			If the extension is standalone (not bundled with other products), the extension convenience build (non-SMPE) MUST package itself into zip, tar, or pax format.
8	v2	x			The extension MUST package a manifest.yaml (or manifest.yml, or manifest.json) into final package.
9	v2	x			If the extension is standalone (not bundled with other products), the manifest file MUST be placed in the root directory after it's extracted or installed.
10	v2	x			The extension manifest MUST contain "name" definition.
11	v2	x			If the extension manifest contains definition of "appfwPlugins", it MUST also define "id" definition.
12	v2		x		It's recommended to define "version" in the extension manifest.
13	v2		x		It's recommended to define "license" in the extension manifest .
14	v2		x		It's recommended to define "title" and "description" in the extension manifest.
15	v2		x		It's recommended to define "build" in the extension manifest to describe the source code information from version control systems.
16	v2		x		The extension is suggested to define "commands.install", "commands.configure", and/or "commands.start" to instruct Zowe how to install, configure or start in Zowe context.
17	v2		x		If the extension contains services that will be registered under API-ML, those services MUST be defined as entries of "apimlServices" in the extension manifest.
18	v2		x		If the extension contains Zowe App Framework plugin(s), these plugins MUST to defined as entries of "appfwPlugins" in the extension manifest.
19	v2		x		If the extension contains Zowe ZIS plugin(s), these plugins MUST to defined as entries of "zisPlugins" in the extension manifest.

Web UIs All

Item	Version	Required	Best Practice	Conformant	Criteria
20	v2	x			All Apps must contain an icon image file located somewhere within the /web folder of the plugin and the plugin definition must specify the location relative to the /web folder as the webContent.launchDefinition.imageSrc property

Web UI IFrame

Item	Version	Required	Best Practice	Conformant	Criteria
21	v2	x			IFrame Apps (apps with framework type "iframe") which embed a top-level iframe within (example) must use the ID "zluxIframe" for that element. This is required for the app to be a recipient of app to app communication.
22	v2	x			Zowe resources must be accessed via the iframe-adapter located within zlux-app-manager/bootstrap/web. Use of window.parent or window.top to access the ZoweZLUX object is non-permissible.

Web UI Non-IFrame

Item	Version	Required	Best Practice	Conformant	Criteria
23	v2	x			DOM elements originating from your App should always be a child of the Zowe viewport DOM element, "com-rs-mvd-viewport"
24	v2	x			Network requests to the Zowe App Server must never be done without the use of the URI Broker
25	v2		x		Access to resources outside the App Server should also be made through the URI Broker whenever possible
26	v2	x			Apps must not pollute the global namespace with regards to Javascript, HTML, and CSS however it is permitted to have 1 Javascript global who's name is equal to the plugin identifier
27	v2	x			When using a library present in the Zowe App Framework core, you must depend on the same version, as seen in here
28	v2		x		Web apps should extend the framework's default build scripts for webpack and typescript.

UI Design

Item	Version	Required	Best Practice	Conformant	Criteria
29	v2		x		Apps should follow the UI Design guidelines at this page

Localization and Internationalization (l10n and l18n)

Item	Version	Required	Best Practice	Conformant	Criteria
30	v2	x			If supporting more than one language, the active language to be used for string selection must be retrieved using the get methods in ZoweZLUX.globalization, which determine language by multiple factors
31	v2		x		If supporting more than one language, no strings visible in a UI should be hard-coded, rather resource strings must be used in accordance with one of the existing internationalization support mechanisms

App Server

Item	Version	Required	Best Practice	Conformant	Criteria
32	v2	x			Data services should be written such that all synchronous and asynchronous errors are caught. Utilize try-catch and check the existence of error objects from asynchronous calls. Uncaught exceptions effect server responsiveness and disrupt clients

Documentation

Item	Version	Required	Best Practice	Conformant	Criteria
33	v2	x			Every HTTP API which is intended for 3rd party use must be documented in swagger 2.0. The swagger document must be stored in doc/swagger
34	v2		x		In addition, it is recommended to have documentation about the format of any Websocket APIs, to be placed within doc

Logging

Item	Version	Required	Best Practice	Conformant	Criteria
35	v2	x			An Apps non-IFrame web components, or App Framework dataservices (eg Javascript and Typescript) must log only through the "zlux" logger
36	v2	x			ZSS services log only through the Zowe ZSS Logger
37	v2	x			Passwords must never be logged

Item	Version	Required	Best Practice	Conformant	Criteria
38	v2		x		Error reporting should follow the standard tooling

Encoding

Item	Version	Required	Best Practice	Conformant	Criteria
39	v2	x			If you want your Apps to work with z/OS Node.js version 12 or greater, all application files must be tagged according to their content type

Storage

Item	Version	Required	Best Practice	Conformant	Criteria
40	v2	x			For persistent storage, user preferences (if applicable to a plugin) must be stored through the configuration data service
41	v2	x			Regarding persistent storage for other plugin storage needs, storing data outside of the configuration dataservice is permitted only within /workspaceDirectory/app-server or /workspaceDirectory/app-server/pluginStatic with a top-level folder equal to their plugin ID. Plugins must not store information anywhere else in any Zowe directories such as /workspaceDirectory or $ROOT_DIR in order to prevent conflict with future Zowe versions and other plugins
42	v2		x		It is advisable for the storage of user preferences to use environment variables for locating directories. Use of the workspace directory environment variable is not required, but should be considered to subvert the use of hard-coded paths
43	v2	x			To be HA compatible, you should use some or all of the storage APIs of Zowe which include but are not limited to the caching service of the API ML, storage API of the App server, and storage API of ZSS

Directory and File Ownership Permissions

Item	Version	Required	Best Practice	Conformant	Criteria
44	v2	x			A conformant application must not modify the contents of the Zowe runtime USS directory and it must not change any directory or file permissions or ownership
45	v2		x		A conformant application must not modify the permissions or ownership of a Zowe workspace directory

Lifecycling as a Zowe address space

Item	Version	Required	Best Practice	Conformant	Criteria
					If the service should be lifecycled by Zowe, then:
46	v2	x			it should provide and specify in its Component manifest a start.sh script
47	v2		x		a validate.sh script
48	v2		x		a configure.sh script
49	v2	x			If the service introduces new environment variables to the zowe.yaml file, these should be prefixed by the provider ID to avoid collisions

Support

Item	Version	Required	Best Practice	Conformant	Criteria
50	v2	x			Submitter describes how Support is provided and Support details are clearly documented

Zowe Explorer for Visual Studio Code

General Extension

Item	Version	Required	Best Practice	Conformant	Criteria
1	v2	x			**Naming:** If the extension uses the word "Zowe" in its name, it abides by [The Linux Foundation Trademark Usage Guidelines and Branding Guidelines](#) to ensure the word Zowe is used in a way intended by the Zowe community.
2	v2		x		**No Zowe CLI plugin installation requirement:** If the extender makes use of a Zowe CLI profile other than the Zowe Explorer default `zosmf` then the extension must not make any assumptions that a matching Zowe CLI plugin has been installed in the Zowe Explorer user's environment.
3	v2	x			**Publication tag:** If the extension is published in a public catalog or marketplace such as Npmjs, Open-VSX, or VS Code Marketplace, it uses the tag or keyword "Zowe" so it can be found when searching for Zowe and be listed with other Zowe offerings.
4	v2	x			**Support:** Extension has documentation with instructions on how to report problems that are related to the extension and not Zowe Explorer. It needs to explain how users can determine if a problem is related to the extension or Zowe Explorer.
5	v2		x		**User settings consistency:** Extender provides a consistent user settings experience. For VS Code extensions, extender follows the recommended naming convention for configuration settings as described in VS Code's configuration contribution documentation
6	v2	x			**Using .zowe in user settings:** Extender avoids starting setting names with the prefix `zowe.`, which is reserved for Zowe Explorer and and other extensions maintained by the Zowe Project .
7	v2		x		**Error message consistency:** Extension follows the recommended error message format indicated in the Zowe Explorer extensibility documentation to provide a consistent user experience with Zowe Explorer.
8	v2		x		**Zowe SDK usage:** Extension utilizes the available Zowe SDKs that standardize z/OS interactions as well as other common capabilities that are used by many other Zowe extensions and tools unless the extension's goal is to provide a new implementation with clearly stated goals.
9	v2	x			**Sharing of profiles with Zowe CLI:** Extensions that utilize Zowe CLI profiles must keep the shared profile instances compatible between Zowe CLI and the Zowe Explorer extension that utilize them.
10	Mark (a) (b) (c)				Extension uses the extensibility APIs provided by Zowe Explorer. Supported methods include (select all that apply a, b, c):
	v2	-	-		a. Extension Accessing Profiles
	v2	-	-		b. Data Provider Extension
	v2	-	-		c. Extension Adding Menus

Extension Accessing Profiles

Criteria for VS Code extensions that want to access the same Zowe CLI profiles that Zowe Explorer uses:

Item	Version	Required	Best Practice	Conformant	Criteria
11	v2	x			**VS Code extension dependency:** If the extension calls the Zowe Explorer API it must declare Zowe Explorer as a VS Code extension dependency by including an `extensionDependencies` entry for Zowe Explorer in its package.json file. This ensures Zowe Explorer and Zowe Explorer API are activated and initialized for proper use by its extenders.
12	v2	x			**Zowe Extender access:** Extension accesses the shared Zowe Explorer profiles cache via `ZoweExplorerApi.IApiRegisterClient.getExplorerExtenderApi()` API as documented in the Zowe Explorer extensibility documentation.
13	v2	x			**Added Profile Type initialization:** If the extension has a dependency on a new Zowe CLI profile type other than the Zowe Explorer default `zosmf`, it is calling the `ZoweExplorerApi.IApiRegisterClient.getExplorerExtenderApi().initialize(profileTypeName)` to ensure that the profile type is supported and managed by the extension without a Zowe CLI plugin installed.
14	v2		x		**Base Profile and Tokens:** Extension supports base profiles and tokens
15	v2	x			**Team Configuration File:** Extension supports the Zowe CLI 7 team configuration file format.

Item	Version	Required	Best Practice	Conformant	Criteria
16	v2		x		**v1 Profile Support:** Extension has a backwards compatibility and it is able to support v1 type of profiles.

Data Provider Extension

Criteria for VS Code extensions that extend the Zowe Explorer MVS, USS, or JES tree views to use alternative z/OS interaction protocols such as FTP or a REST API.

Item	Version	Required	Best Practice	Conformant	Criteria
17	v2	x			**VS Code extension dependency:** If the extension calls the Zowe Explorer API it must declare Zowe Explorer as a VS Code extension dependency by including an `extensionDependencies` entry for Zowe Explorer in its package.json file. This ensures Zowe Explorer and Zowe Explorer API are activated and initialized for proper use by its extenders.
18	v2	x			**New Zowe CLI profile type:** Extension registers its new API instances with a new profile type name for the different Zowe Explorer views via the `ZoweExplorerApi.IApiRegisterClient.register{Mvs|Uss|Jes}Api(profileTypeName)` call as indicated from the Zowe Explorer extensibility documentation
19	v2		x		**Matching Zowe CLI Plugin:** Provide a Zowe CLI Plugin for the data provider's new profile type that implements the core capabilities required for the new protocol that users can then also use to interact with the protocol outside of the Zowe Explorer extension using Zowe CLI commands.
20	v2	x			**Data provider API implementation:** Extension fully implements and registers to at least one of the three Zowe Explorer interfaces or alternatively throw exceptions that provide meaningful error messages to the end-user in the 'Error.message' property that will be displayed in a dialog.
21	v2		x		**API test suite implementation:** If the extension implements a Zowe Explorer API data provider interface, it should implement a test suite that calls each of the implemented API methods.

Extension Adding Menus

Criteria for VS Code extensions adding menu and commands to VS Code that utilize Zowe Explorer data or extend Zowe Explorer capabilities.

Item	Version	Required	Best Practice	Conformant	Criteria
22	v2		x		**VS Code extension dependency:** If the extension calls the Zowe Explorer API it must declare Zowe Explorer as a VS Code extension dependency by including an `extensionDependencies` entry for Zowe Explorer in its package.json file. This ensures Zowe Explorer and Zowe Explorer API are activated and initialized for proper use by its extenders.
23	v2	x			**Command operations:** If the extension is adding new commands to Zowe Explorer's tree views, the commands must not replace any existing Zowe Explorer commands.
24	v2		x		**Command categories 1:** If the extension adds to contributes.commands in package.json, the value assigned to the category property contains the extension name.
25	v2	x			**Command categories 2:** If the extension assigns values to the category property in contributes.commands in package.json, the value cannot be "Zowe Explorer".
26	v2	x			**Context menu groups:** If contributing commands to Zowe Explorer's context menus, the extension follows the Zowe Explorer extensibility documentation and adds them in new context menu groups that are located below Zowe Explorer's existing context menu groups in the user interface.
27	v2		x		**Adding New Menu Items:** If the extension is adding new commands and context menu entries to the Zowe Explorer tree view nodes, the new command name is consistent with the terminology and naming conventions of the existing Zowe Explorer menu entries. More information is provided in the Zowe Explorer extensibility documentation.
28	v2	x			**Existing Menu Items:** Extension does not overwrite any existing Zowe Explorer command and context menu entries

Item	Version	Required	Best Practice	Conformant	Criteria
29	v2	x			**Context menu items:** If contributing commands to Zowe Explorer's views (such as Data Sets, USS, or Jobs), the extension should only add them to the view's right-click context menus.