**Local Applications Guide**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#LocalApplicationsGuide)

This is a working draft to establish guidelines and reference for Local Applications Development for <the new thing that is not the original AWIPS>

Please add new sections as needed. This document should define the rules and should be kept as short and concise as possible. Subpages that are part of the rules should be named using [CamelCase](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/CamelCase) with the prefix LocalApplicationsGuide (i.e., LocalApplicationsGuidePlugins for rules for plugins). Wiki pages with this name prefix will be part of the official policy document. Procedural instructions, code examples, etc. should be links to other pages named without the LocalApplicationsGuide prefix. These are reference documents, not part of the policy.

**Local Applications Guide**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#LocalApplicationsGuide)

(for governance issues still being worked, see the [local apps process](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsProcess) page)

**Local Applications Environment**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#LocalApplicationsEnvironment)

This section describes the software tools, the software development environment, and the runtime environment. It gives guidelines for setting up a local applications environment which protects the baseline AWIPS data and software configurations.

**Directory Structure**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#DirectoryStructure)

The following defines the directory structure for local applications. This structure is for applications that are not required to follow different installation or interface procedures like CAVE and edex plugins, uEngine Web scripts, etc. Please see the Plugins section for specifics for CAVE and edex plugins.

|  |  |  |
| --- | --- | --- |
| **Directory** | **Environment Variable** | **Description** |
| /localapps | LOCALAPPS\_HOME | Top Level directory |
| $LOCALAPPS\_HOME/runtime | LOCALAPPS\_RUN | Path from where all local applications will be run. |
| $LOCALAPPS\_HOME/dev | LOCALAPPS\_DEV | Top level directory for software development. |
| $LOCALAPPS\_HOME/lib/<language> | LOCALAPPS\_LIB\_<language> | Common libraries/API's for use by more than one application. There will be separate subdirectories for different programing languages such as python, tcl, etc. |
| $LOCALAPPS\_HOME/logs | LOCALAPPS\_LOGS | Common log directory |

Each application should be developed and distributed in its own application named directory. In general, programs will be installed in $LOCALAPPS\_RUN/appname with the following structure:

|  |  |
| --- | --- |
| **Subdirectory** | **Description** |
| bin | Executable files |
| config | Default/sample configuration files |
| data | Application runtime data |
| doc | Documentation files |
| etc | Site configured files |
| lib | Application libraries |
| log | Application log files |

Notes:

1. This structure requires the application be developed with the same structure in $LOCALAPPS\_DEV/appname
2. Subdirectories data and logs may be symbolic links to point to common locations like $LOCALAPPS\_LOGS.
3. Applications may be organized into a hierarchical directory structure within $LOCALAPPS\_DEV and $LOCALAPPS\_RUN; they are not required to be directly beneath the top level dev and runtime directories. For example, a regionally required application should be installed in $LOCALAPPS\_RUN/<region>/appname.

**Environment Variables**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#EnvironmentVariables)

Applications should avoid using hard coded paths and use system environment variables instead.

|  |  |
| --- | --- |
| **Directory** | **Environment Variable** |
| /localapps | LOCALAPPS\_HOME |
| $LOCALAPPS\_HOME/runtime | LOCALAPPS\_RUN |
| $LOCALAPPS\_HOME/dev | LOCALAPPS\_DEV |
| $LOCALAPPS\_HOME/lib/java | LOCALAPPS\_LIB\_java |
| $LOCALAPPS\_HOME/lib/javascript | LOCALAPPS\_LIB\_javascript |
| $LOCALAPPS\_HOME/lib/perl | LOCALAPPS\_LIB\_perl |
| $LOCALAPPS\_HOME/lib/python | LOCALAPPS\_LIB\_python |
| $LOCALAPPS\_HOME/lib/shell | LOCALAPPS\_LIB\_shell |
| $LOCALAPPS\_HOME/lib/tcl | LOCALAPPS\_LIB\_tcl |
| $LOCALAPPS\_HOME/logs | LOCALAPPS\_LOGS |

**Environment Variables set by scripts**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#EnvironmentVariablessetbyscripts)

AWIPS2 uses scripts to setup the environment variables to run rehosted AWIPS applications. The following section shows these scripts and the variables being set:

**Hydro Related Variables**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#HydroRelatedVariables)

The /awips2/edex/data/hdf5/hydroapps/set\_hydro\_env script is used to set the parameters required to access the hydro database and applications. The following are a few of the variables being set by this script and replace the calls to the get\_apps\_default utility:

* PGHOST (get\_apps\_default pghost)
* PGPORT (get\_apps\_default pgport)
* PGUSER (get\_apps\_default pguser)
* DB\_NAME (get\_apps\_default db\_name)

**GFE Smart Tools**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#GFESmartTools)

Smart Tools, Procedures, and Utilities should run in a similar manner as in AWIPS.

GFE files in the repository should be named "name.class" where name and class are the values used with the -n and -c arguments to ifpServerText. SmartInits should use a .py extension. GFE files should be installed using the ifpServerText program and not written directly to the edex file system. The valid ifpServerText classes are:

* EditAreaGroup
* EditArea
* WeatherElementGroup
* ColorTable
* SelectTR
* SampleSet
* Tool
* Procedure
* Utility
* TextUtility
* TextProduct
* Config
* Combinations

**Tools**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#Tools)

**Procedures**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#Procedures)

**Utilities**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#Utilities)

**Plugin Development**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#PluginDevelopment)

Plugin development should occur by the field sites after initial deployment with guidance from the development organization and/or experienced java developers from select field sites.

**CAVE Plugins[¶](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide" \l "CAVEPlugins" \o "Link to this section)**

CAVE Plugins should reside in the $CAVE\_HOME/Plugins folder and follow the naming convention of:

gov.noaa.nws.SITE.PLUGINNAME\_VERSION

where SITE is the developer's site and PLUGINNAME\_VERSION is a unique ID for a plugin.

**EDEX Plugins[¶](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide" \l "EDEXPlugins" \o "Link to this section)**

It is not known at this time how EDEX Plugins will be developed.

|  |  |
| --- | --- |
| **Directory** | **Description** |
| $EDEX\_HOME/lib/plugins | Default EDEX Plugin location |

**Event Processing**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#EventProcessing)

**Crons / Quartz Events**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#CronsQuartzEvents)

Heartbeat managed crons will run on dx1/2, px1/2. There will be no heartbeat managed function on dx3/4. Most site cron jobs will likely remain crons. Quart timers can be used for things that need access to edex or need to run uEngine scripts.

**Trigger / Subscription Events**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#TriggerSubscriptionEvents)

ALL [Triggers](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/triggers) will need to be replaced with [subscriptions](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/Subscriptions) Examples of triggers are:

1. LDAD triggers ldadTrigger.template
2. Hydro triggers hydroSiteConfig.txt
3. ADAPPT triggers adaptTriggerAll.template
4. Fax triggers faxTrigger.template
5. Local triggers siteTrigger.template

**Manual / User Events**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#ManualUserEvents)

Manual Events are scripts, programs, or applications that are invoked via command line or Application Launcher on the Workstation. They should conform to the Local Application Environment Directory Structure above.

**Development Environment**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#DevelopmentEnvironment)

1. Cave and edex plugins must be developed using Eclipse and the ADE.
2. All applications should be developed using Subversion version control software and shared local applications are required to be placed into the NCLADT Subversion repository. See [HowToSubversion](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/HowToSubversion).

**Coding Standards and Best Practices**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#CodingStandardsandBestPractices)

1. The primary development language for AWIPS II should be Python. This language will be supported by the NCLADT team. In addition, by using a common language (Python will also be used by the baseline system for smart tools, GFE procedures, and derived parameters), we will be able to develop a standardized library of common functions.
2. For more complex tasks, including CAVE and edex plugins, Java should be used and integrated by developing in the AWIPS II Development Environment (Eclipse ADE).
3. Other languages will also work in the AWIPS II environment; however, they may not have direct APIs into the AWIPS II architecture.
4. It is highly recommended to insert the repository Revision number and date inside all program files. This is basically a one time task when adding a new file. See [SubversionTips](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/SubversionTips) for instructions. This will allow developers and users to unequivocally know what version of a program or file is being used.

**Documentation Standards and Best Practices**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/LocalApplicationsGuide#DocumentationStandardsandBestPractices)

1. Application documentation should be provided in a Web-friendly format such as HTML or PDF. Wiki pages are encouraged to be used for the documentation.
2. Each released version of an application must have Release Notes that clearly identify changes, especially new or changed configuration items.