EAF Command Line Processing

This document explains how and EAF-based application processes command line parameters.

All MFC CWinApp applications start with a call to InitInstance(). CEAFApp::InitInstance() calls the ParseCommandLine() method the a CEAFCommandLineInfo object returned from GetCommandLineInfo().This occurs before any application plug-ins are created. Your application can control application start up and initialization through a custom CEAFCommandLineInfo object. Override GetCommandLineInfo() and return your custom command line information object. During application start up and initialization, refer to your command line information object and alter the behavior of the application as needed.

After the application is started and initialized, the last step in CEAFApp::InitInstance() is to call the polymorphic function CEAFApp::ProcessCommandLineOptions().

ProcessCommandLineOptions() first passes the application command line information to the MFC function ProcessShellCommand(). If a filename was provided on the command line, MFC will attempt to find the associated document template, create the document object, and open the file.

Next, ProcessCommandLineOptions() finds the Application Plug-in object that registered the document template for the open document. IEAFAppPlugin::ProcessCommandLineOptions() will be called on the application plug-in object so that it can process any special parameters.

If a document was not opened, ProcessCommandLineOptions() iterates through all the Application Plug-in objects until one of them handles the command line parameters. If the command line parameters are not handled, an error occurs and the application closes after displaying an error message.

Each implementation of IEAFAppPlugin must decide if further command line processing is necessary. If so, the plug-in can implement the application specific procedure. A default procedure is provided by calling ProcessCommandLineOptions() on the open document. By default, CEAFDocument-based documents call ProcessCommandLineOptions() on all of the installed and activated document level plug-ins. If the document is CEAFBrokerDocument-based, then ProcessCommandLineOptions() is called on the agents implementing the IEAFProcessCommandLine interface.

When implementing ProcessCommandLineOptions(), it is important to know that the CEAFCommandLineInfo object that is passed as a parameter is the object that parsed the command line parameters in CEAFApp::InitInstance(). This object does not contain your command line parameters. You will need to re-parse the command line using your custom CEAFCommandLineInfo object. After parsing the command line parameters, assign the results of your object to the one passed as a parameter. Object slicing will occur, however it is important to set the m\_bError, m\_strErrorMsg, and m\_bCommandLineMode parameters correctly.

If invalid command line parameters were supplied by the user, you can customize the error and usage message by setting the m\_strErrorMsg data member of calling SetErrorInfo(). This message text is displayed to the user when an error occurs.

The PGSuperAppPlugin object provides an example of processing custom command line parameters. The TxDOTAgent object provides an example of an extension agent, extending the command line options.