Common Component Standard Proposal

Date	7.21.2015
Spec Title	Configuration Manager
Author	mjacobs@progress.com
Status	New
Version	1.0

Executive Summary

This document contains a proposal to the Common Component Standards (CCS) Steering Committee to form a new project team whose goal it is to produce a specification for a *Configuration Manager* OERA component. The following sections of the document describe in general terms what the *Configuration Manager* component is, the benefits it provides to application developers, and what the team's deliverables would be.

Component Description

A *Configuration Manager* component would be a member of the OERA common services group and provide common API READ access to a diverse set of configuration data sources used by one or more other application subsystems or OERA common components. Composed of three interfaces, the *Configuration Manager* would provide the application or component developer the ability to:

1. Load configuration data of a certain organization and format from some external source
2. Obtain a subset of configuration property data for a specific application subsystem or component
3. Provide read-only access to individual configuration property value(s)

The *Configuration Manager's* interfaces would effectively produce a single methodology by which external configuration data could be consumed from one or more physical sources of varying types and locations. This single methodology would afford the developer with a consist programming model whereby they would:

1. Instantiate an instance of a *Configuration Manager* implementation
2. Use the *Configuration Manager* instance to open and load an external source of configuration data via a Universal Resource Locator

3. Use the *Configuration Manager* instance to return a filtered collection [instance] of the loaded configuration data that pertains to a specific subsystem or component
4. Use the collection instance to access individual configuration property value, or values, on demand

The *Configuration Manager's* APIs would facilitate an application's access to configuration data via common single or multi-value retrieval techniques commonly found in common configuration data storage formats.

All of the *Configuration Manager's* abstraction of physical storage serves the end goal of writing an application's code once without regard to where and how configuration data is physically stored and formatted. For example, a *Configuration Manager* component may, depending upon the standards implementer, offer access to:

- Flat file name-value records
- Java system property files using hierarchical name spaces
- Windows INI files using groups and nested groups
- LDAP / Active Directory
- Windows registry hives
- Database storage
- …

A Configuration Manager component's interfaces would be defined as OOABL interface artifacts and require the minimum functionality provided by the OpenEdge 11.3.0 release. While the end result of the standard may be supportable in earlier OpenEdge releases it is not a requirement that it do so.

Benefits and Use Cases

Using externally managed configuration data is a foundational part of any application, subsystem, and/or component. As applications have become more complex so has the ability to access the various locations where developers have chosen to store configuration data. The benefit a *Configuration Manager* component brings to application development is to lessen the impact of working with a diverse set of configuration data storage and formats. Those benefits manifest themselves in two primary aspects.

First, the business application developer may employ a *Configuration Manager* as a single method of accessing configuration data without embedding the technologies needed to access different physical sources.

Second, when employing other common component implementations that also utilize a *Configuration Manager*, they too become isolated from physical storage concerns and may dynamically change to operate in conjunction with the business application's configuration storage.

Related / Dependent Common Component Specifications

The *Configuration Manager* component is not dependent upon any other common component standard specification. However, the *Configuration Manager* component specification may be incorporated into any other common component standard specification.

Project Team Requirements

A *Configuration Manager* team has not yet been formed. The team's formation will commence upon acceptance of this standards specification proposal by the CCS Steering Committee. The composition of the team will be supplied to the CCS Steering Committee within 30 calendar days after acceptance.

The *Configuration Manager* team will provide regular updates whereby the CCS Steering Committee may gauge their progress, offer guidance, or require additional information.

The team's goal would be to produce a draft *Configuration Manager* component specification within a period of 90 days after the team's formation was complete. Upon initial acceptance by the CCS Steering Committee for meeting specification requirements, the specification would be published to the CCS community for comments for a period of 90 days. During the 90 day community comments period the team is responsible for answering all comments and requests for clarification. At the completion of the 90 day comments period a second version of the *Component Manager* specification will be produced that incorporates those public comments accepted by the team. The second version of the *Configuration Manager* specification will then be ready for a formal review by the CCS Steering Committee for acceptance, cancelation, postponement, or denial.

An essential part of the project team's delivery is that of an operating reference sample that may assist is the community's in gaining a better viewpoint of how a Configuration Manager component may be used, rather than just reading a specification. This sample would also serve as a physical illustration of the originating team's initial vision of the standard.

Upon acceptance of the Configuration Manager team will have one last task, and that is to publicly publish the standard, sample, and any other artifacts generated as part of the project.

The team would remain until one of those three outcomes is reached: acceptance, cancelation or postponement. If the CCS Steering Committee postpones the project the team will be disbanded and should the project be resumed at a later date a new team would be formed.