Technical Specifications for Software Application Development and Acquisition

Commission for Environmental Cooperation
Revised 5 October 2012

Prepared by Jean-François Pilon, Manager, Information Services
# Table of Contents

1. CEC APPLICATION DEVELOPMENT / GENERAL SPECIFICATIONS 3
   1.1 Languages 3
   1.2 Database Compatibility 3
   1.3 Database Basics 3
   1.4 Maintenance and Archiving Plans 4
   1.5 Database Structure and Consistency 4
   1.6 Quality Assurance 4

2. APPLICATION AND DATA ARCHITECTURE 4
   2.1 Internet Application 5
      2.1.1 Static Web Pages 5
   2.2 Software Applications 8
   2.3 Graphics and Design 8
   2.4 Manage Human Language 8
   2.5 Character Encoding 9
   2.6 Application Delivery Document Outline 9
   2.7 Support During the Development 9

3. APPLICATION DELIVERY DOCUMENT 10
   3.1 Introduction 10
      3.1.1 Purpose and Scope 10
      3.1.2 System Overview 11
      3.1.3 Project References 11
      3.1.4 Glossary 11
   3.2 Management Overview 11
      3.2.1 Description of Integration 11
      3.2.2 Responsibilities 11
      3.2.3 Activities and Tasks 12
   3.3 Integration Support 12
      3.3.1 Resources and Their Allocation 12
      3.3.2 Training 12
      3.3.3 Testing 12

This document addresses software and Web application development or purchase by the CEC and provides guidelines to CEC consultants who are required to develop such applications, as well as to CEC Secretariat staff who request purchasing new applications.
1. CEC APPLICATION DEVELOPMENT / GENERAL SPECIFICATIONS

1.1 Languages
The CEC operates in three official languages and delivers information to the public in each language. These are: English, Spanish and French. Any application designated for the public should, where possible, include the flexibility of delivering information in the three languages (e.g., including three text fields per type in the database architecture).

1.2 Database Compatibility
The CEC accepts two technologies for database architecture:

- **Microsoft SQL Server.** Microsoft SQL Server is a relational database management system produced by Microsoft. It supports a superset of Structured Query Language (SQL), the most common database language. It is commonly used by businesses for small- to medium-sized databases and by large enterprise databases and competes with other relational database products for this market segment. The CEC strongly recommends this Microsoft SQL Server as the primary choice for Web applications. We currently use MS SQL 2005.

- **MySQL.** MySQL is a popular choice of database for use in Web applications, and is a central component of the widely used LAMP open-source Web application software stack—LAMP is an acronym for “Linux, Apache, MySQL, Perl/PHP/Python.” MySQL is an open-source database management system and is used in some of the most frequently visited websites on the Internet. Our server currently has version 5.0.95.

It is the responsibility of the author to ensure that the database format is compatible with the CEC operating system and, for databases intended for online availability, the Web interface. The author must seek prior, written approval from the CEC to use a technology other than those listed above.

The consultant is responsible for making sure that every trigger, stored procedure and special database architecture is described in the Application Delivery Document (see chapter 3).

1.3 Database Basics
All databases need full documentation and transparent quality assurance. At a minimum, the CEC requires the following:

- Proper file-naming (CEC_project reference.extension).
- A clearly marked title of the dataset, database or information service.
- The database category (project or ongoing). These categories are defined in Template 3 on page 24 of the CEC’s **Quality Assurance Policy and Procedures (QAPP).**
- The file name and narrative description (including title, date, time period, geographic specifications, acronyms, and field and type descriptions) in a separate document, to be submitted with the dataset, database or information service. If the database uses data from a previous publication or is an ongoing database, a historical record of the database must also be included in the narrative description.
- Database architect and access information (e.g., name of individual/position within the organization responsible for the data, passwords, and authority to edit, change,
1.4 Maintenance and Archiving Plans

The dataset and database architecture must be designed according to updating needs (section 1.3 above defines different database categories and their respective updating needs). Particularly if the project is ongoing, it is important to:

- build data capacity flexibility into the architecture to provide for expanded data sets and languages; and
- select software and technology to support long-term database updating, development and maintenance needs.

Clear documentation of how the database was built must be recorded to ensure traceability and knowledge transfer capability. Lastly, a record of metadata should be submitted along with the dataset or database, for CEC archiving purposes.

1.5 Database Structure and Consistency

The CEC generates and stores large amounts of data for North America. To the extent possible, the information products commissioned or contracted by the CEC should be consistent with previous work, to provide for cross-connection of datasets. The data classification system must be designed for coherency within the context of CEC programming and past work. This also applies to nomenclature, definition of terms, labels, abbreviations, acronyms, and units (including number format, decimal numbers, rounding, etc.). Wherever possible, datasets should be organized in fully documented relational databases. Data must be checked for consistency and compatibility; the author should verify that measurements, numerical data, units and definitions are consistent across sampling locations and with previous work. Validation criteria (e.g., field properties, validation rules, and input masks) should be implemented to ensure data are entered correctly.

1.6 Quality Assurance

Data quality is addressed in the CEC’s Quality Assurance Policy and Procedures (QAPP), specifically section 2.3: “Data and Information Management Procedures” (see <www.cec.org/Storage/82/7706_QAPP_en.pdf>). The author is expected to read the procedures specified in QAPP and to follow the Data and Information Quality Assurance Plan (DIQAP) completed by the responsible project manager in the early stages of the project planning process.

The CEC applies quality assurance throughout all stages, from data assembly through data analysis and use to archiving. It is the responsibility of the author to seek knowledge of end-uses that may result from the project (e.g., digital information products such as online databases, downloadable datasets, query services, or GIS maps) and plan the work to include controls at each step of the project.

2. APPLICATION AND DATA ARCHITECTURE

The consultant is responsible for creating the design and architecture for the application he is developing. This step should proceed at the beginning of the project. In some cases, the CEC can request the consultant to present his architecture and/or design plan in order to help us analyze and prepare for the integration and deployment of his solution.

The consultant can receive some support from the CEC’s Web Development and Information Services department in order to help with the development and understanding of the specifications.
2.1 Internet Application

2.1.1 Static Web Pages

Static Web pages consist of text, images, and HTML formatting tags. These pages are created manually and maintained so that when information changes, so must the page. This usually involves loading the page into an editor which makes the changes, reformatting text if needed and then saving the file. Not everyone in the organization can make these changes. The webmaster or Web design team is responsible for maintaining the site and implementing all changes and enhancements. This means that information is often out of date when it finally makes it onto the website.

Please also see the Graphics and Design section (2.3) in order to be familiar with CEC graphics and design specifications.

2.1.1.1 Dynamic Web Pages

Dynamic Web pages contain very little text. Instead, they pull the needed information from other applications or databases.

Web Programming

If the contract stipulates that a specific programming language should be used to develop the application, the consultant should respect it. Otherwise, the consultant can choose from the following list to develop the application needed by the CEC. The following languages are compatible with the technologies used by the CEC:

- **ASP.** Active Server Pages (ASP) is a technology that allows you to dynamically generate browser-neutral content, using server-side scripting. The code for this scripting can be written in any of several languages and is embedded in special tags within the otherwise-normal HTML code making up a content page. This heterogeneous scripting/content page is interpreted by the Web server only upon the client's request for the content.

  Note: The CEC is not using ASP as the primary Web technology to operate its applications.

- **.NET.** The .NET Framework sits on top of the operating system, which can be any flavor of Windows*, and consists of a number of components. Currently, the .NET Framework consists of the following:
  1. Four official languages: C#, Visual Basic .NET, Managed C++, and Jscript .NET.
  2. The Common Language Runtime (CLR), an object-oriented platform for Windows and Web development that all these languages share.
  3. A number of related class libraries, collectively known as the Framework Class Library (FCL).

* Because of the architecture of the Common Language Runtime, the operating system can be potentially any flavor of Unix or any other operating system.
**Note:** The CEC is not using the .NET as the primary Web technology to operate its applications. The CEC operates the .NET Framework version 2.0 on its Web servers.

- **PHP.** PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. It is part of the ubiquitous LAMP stack. Our server uses PHP 5.2.10.

- **ColdFusion.** ColdFusion is an application server—a piece of software that (usually) resides on the same computer as the Web server, enabling it to do things it wouldn’t normally know how to do. To learn more about this technology, please visit [http://www.macromedia.com/coldfusion](http://www.macromedia.com/coldfusion).  

  **Note:** The CEC is using ColdFusion MX 7 technology. The CEC will accept CFC and CustomTags, according to their complete full description in the Application Delivery Document (see chapter 3).

- **HTML.** HTML is the *lingua franca* for publishing hypertext on the World Wide Web. It is a non-proprietary format based upon SGML (Standard Generalized Markup Language). HTML should only define the structure and outline of a document and offer a structured content. It is not intended to define the look and feel of the content on the page beyond rudimentary concepts (headers, paragraphs, and lists). The presentation attributes of HTML have all been deprecated and style should be contained in style sheets.

- **CSS (Cascading Style Sheets).** The second component of a Web page is the presentation information contained in the Cascading Style Sheet (CSS.) Web browsers’ successful implementation of CSS has given a whole generation of Web authors site-wide control over the look and feel of their websites.

  Just as the information on a Web page is semantically described in the HTML Markup, CSS describes all presentation aspects of the page via a description of its visual properties. CSS is powerful in that these properties are mixed and matched via identifiers to control the page’s layout and visual characteristics through the layering of style rules (the “cascade”). You can use the CSS Validation Service from the World Wide Web Consortium (W3C) organization (<http://jigsaw.w3.org/css-validator/>).

  The consultant can request the CEC’s IT department to get the CSS standard template used throughout our Web applications.

- **JavaScript / JQuery.** JavaScript is the third major component of most Web pages. JavaScript code, when properly applied to a Web page, enhances the overall user experience through events and controlling the behaviour layers.

  JavaScript has seen an increase in popularity recently as new browser versions now permit the creation of browser-based Web applications. We can now make Web pages act more and more like any other application by manipulating in real time the visual elements, without full Web page reloads. And by using the AJAX techniques, the data themselves can be loaded as needed.
The CEC recommends using JavaScript as the de facto standard for developing rich internet application.

- **Java applets.** These are small programs that run within the Web browser (actually, they run within a Java Virtual Machine, but we won’t worry about that just yet). Applets were popular in the late 1990s but are seldom used now because they are hard to write, slow to download, and often incompatible with all the computers, operating systems, and browsers in use. The CEC strongly recommends using an alternative.

- **Macromedia Flash.** Flash provides a mechanism for creating rich and portable interactive user interfaces (complete with audio, video, and animation, if needed). As Flash is not supported thoroughly on the new mobile platforms, we strongly recommend using an alternative.

- **XML.** Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. The CEC strongly recommends using the Validator for XML Schema from the World Wide Web Consortium (W3C) (<http://www.w3.org/2001/03/webdata/xsv>).

- **XSL.** The Extensible Stylesheet Language (XSL) is a family of recommendations for defining XML document transformation and presentation. It consists of three parts:
  1. XSL Transformation (XSLT)—a language for transforming XML.
  2. The XML Path Language (XPath)—an expression language used by XSLT to access or refer to parts of an XML document. (XPath is also used by the XML Linking specification.)
  3. XSL formatting Object (XSL-FO)—an XML vocabulary for specifying formatting semantics.

- **ActiveX.** A set of properties that define how applications are to share information with each other. These properties are known as controls.

- **DRUPAL.** Drupal is an open-source content management platform powering millions of websites and applications. It’s built, used, and supported by an active and diverse community. It can be used to build everything, from personal blogs to enterprise applications. Thousands of add-on modules and designs let you build any site you can imagine. Distributions are a collection of pre-configured themes and modules for feature-rich websites, giving you a head start. We currently support Drupal 6.

For all code languages, we require indentation to be done by using the space character. Hitting Tab in your editor shall be equivalent to four spaces. The CEC prefers readability over file-size savings, for easier maintenance. Plenty of whitespace is encouraged.

All application must be documented in such detail that any other programmer can understand what has been developed at a technical level. The consultant will have to technically describe all functions and procedures involved in the developed application. These details should be retrieved from the Application Delivery Document, as well as in commented form in the code itself. Ideally, you should use the commented code to build your documentation, using tools like Natural Docs.
2.2 Software Applications
When the consultant needs to develop an application shipped as software, every specification will be mentioned in the contracts under “Terms of References.”

2.3 Graphics and Design
The consultant needs to respect, at all times, the possibility that the CEC may need to modify the appearance of the information on the screen. This should be done by simple methods:

- The use of a centralized CSS file.
- The use of simple graphics/images, keeping in mind that they can be modified at any time.
- Keeping the appearance simple so it can be integrated easily into one of the standard CEC templates.

The consultant can receive some support from the Web Design team in order to help with the appearance of the application. The CEC’s IT department can provide the consultant with a copy of the CEC CSS file, if needed.

2.4 Manage Human Language
The CEC operates in three (3) official languages when it is time to deliver information to any kind of public. These languages are:

- English
- Spanish
- French

Any application designated for the public should consider the possibility to deliver information in the three (3) languages. This means including three text fields per type in the database architecture.

If the consultant is developing the application using the dynamic method, the following should help to be fully compatible with the CEC’s methods:

- Use a parameter within the URL to indicate the language with the variable SiteLanguageID. This method entails that all instruction for the three languages are on the same page. The page will always operate through the parameter passed through the URL but the queries and dynamic method included on the code-page-code will not replicate the page by language like the third method proposed. Keep in mind that this method is open to SQL injection to a certain point, and some measures should be taken in the code to avoid database intrusions.

Here is an example from the CEC corporate website, in English, French and Spanish respectively:

- http://www.cec.org/Page.asp?PageID=1115&BL_WebsiteID=1&AA_SiteLanguageID=1
• Use a session variable in order to indicate in which language the user should navigate through the page. Keep in mind the possibility to modify the language in the same session again.
• Use a single page per language. This method is often used when dealing with static pages.

For example, a page describing an event will exist in the system as three pages, each in a different one of the three languages.

If the consultant needs to use another method to dynamically operate the navigation between the three languages, he/she should have to describe the method in the Application Delivery Document (see section 2.6 below).

2.5 Character Encoding
Character encoding specifies mappings from a character set to the integer numbers that represent the characters on a computer. The CEC recommends that you encode all international application pages as iso-8859-1.

2.6 Application Delivery Document Outline
Along with the contract, the consultant must fill out the document titled "Application Delivery Document," described in chapter 3. This will act as a bridge of information between the consultant and the IT department and will enable us to understand every technical detail of the developed application. This document includes specific questions to be answered by the consultant. The document also includes several descriptive pages, in order to understand the functions of the application and the possibility for the CEC to modify the appearance of the application. This will ensure congruency with the graphics and design corporate standards of the organization.

2.7 Support During the Development
The consultant can request support and confer with the CEC IT department throughout the whole process. In the event that the IT department is not able to respond adequately to the request, it will refer the consultant to another source of support.
3. APPLICATION DELIVERY DOCUMENT

The Application Delivery Document defines the activities necessary to integrate the application and its components into the CEC system(s) and/or server(s). The Application Delivery Document contains an overview of the developed application, a brief description of the major tasks involved in the integration, the overall resources needed to support the integration effort. The plan is developed during the Requirements Phase and is updated during the Design, Implementation, and Verification phases; the final version is provided in that last phase.

It is the responsibility of the Program Manager to fill in this document and to submit it to the Web and Information Services manager of the CEC. The latter may help in formalizing the information needed, as well as may serve as a bridge between the Program Manager and any third party, if need be.

The Application Delivery Document template is described below.

3.1 Introduction

This section provides an overview of the application and includes any additional information that may be appropriate.

3.1.1 Purpose and Scope
This section describes the purpose and scope of the Application Delivery Document. Reference the application name and identify information about the application to be integrated.

3.1.2 System Overview

This section provides a brief overview of the application to be integrated, including a description of the application and its architecture. Describe the environment/infrastructure and how this application will integrate into it. Include any risk involved and the mitigating procedures to reduce or eliminate that risk.

3.1.2.1 Application Description

This section provides an overview of the processes the application is intended to support. If the application is a database or an information system, provide a general discussion of the description of the type of data maintained and the operational sources and uses of those data. Also include all interfaces to other application.

3.1.2.2 Module Description

This section provides an overview of the processes the module is intended to support. If more than one module is being integrated, provide descriptions of each module (or unit) in this section.

It should at least contain the following:

a) Database architecture, tables, triggers, stored procedures
b) ColdFusion CFC and CustomTags full descriptions
c) Technical description
d) Special method to dynamically operate the navigation between the three languages

3.1.3 Project References

This section provides key project references and deliverables that have been produced before this point in the project development.

3.1.4 Glossary

Provide a glossary of all terms and abbreviations used in the document. If it is several pages in length, it may be placed in an appendix.

3.2 Management Overview

The subsequent sections provide a brief description of the integration and major tasks involved in this section.

3.2.1 Description of Integration

This section provides a brief description of the application modules and the integration approach.

3.2.2 Responsibilities

In this section, identify the System Proponent, the name of the responsible or issuing organization, and titles and telephone numbers of the staff who serve as points of contact for
the application integration. It should also include who has approval authority for each module of the application. If this activity is contracted out, list the names and phone numbers of the contractor responsible for the development and integration.

3.2.3 Activities and Tasks

This section provides a brief description of each major task required for the integration of the application. Also include a schedule for when these tasks are expected to be completed. Add as many subsections as necessary to this section to describe all the major tasks adequately. Include the following information for the description of each major task, if appropriate:

a) What the task will accomplish
b) Resources required to accomplish the task
c) Key person(s) responsible for the task
d) Criteria for successful completion of the task

Examples of major tasks include the following:

a) Providing overall planning and coordination for the integration
b) Providing appropriate training for personnel
c) Providing appropriate documentation on each module for integration
d) Providing audit or review reports
e) Documenting software unit and database
f) Establishing software requirements
g) Establishing test procedures
h) Conducting unit testing
i) Conducting qualification testing
j) Integrating modules or application into system(s)/application(s)

3.3 Integration Support

This section describes the support software, materials, equipment, and facilities required for the integration, as well as the personnel requirements and training necessary for the integration.

3.3.1 Resources and Their Allocation

In this section, list all support software, materials, equipment, and facilities required for the integration. Describe the test environment and any resources needed. Describe the number of personnel needed and an estimate of the costs for them, if applicable.

3.3.2 Training

This section addresses the training, if any, necessary to prepare for the integration and maintenance of the system; it does not address user training. If contractors are performing the integration functions and activities, this may not be necessary. If, however, CEC staff are performing these activities, some training might be needed. List the course(s) needed, by title, instructor and cost, where applicable.

3.3.3 Testing
In this section, list all the test requirements for each module. If more than one module is being tested, include a description for each one. Include the descriptions of the data included, procedures for testing, who is responsible for the testing, and a schedule. This could be accomplished in one plan or several, depending on the complexity of the module being tested.

3.3.3.1 Change Procedures and History

Include all changes made during the unit testing.