

CERS EDT Services Manual

Version 1.00 October 3, 2013

This is a technical manual intended for information technology professionals and others who have direct responsibility for implementing machine-to-machine, electronic data transfer (EDT) to the Cal/EPA Unified Program’s California Environmental Reporting System (CERS2). This manual assumes the reader knowledgeable about Unified Program participants, mandates, forms (UPCFs), etc.

The Version 1.00 release of this document updates the 0.91 release from April 2012. Most of the changes involve a dramatic reduction in the size of the document by moving business rules into the new [CERS2 Business Rules](#) document and/or the [EDT Endpoint Documentation](#), along with updating EDT endpoint names to match what they are now called. Cal/EPA anticipates releasing occasional updates to this document to reflect lessons learned and new EDT and CERS UI features that may be relevant to EDT implementers. These releases will be announced via the [EDT Implementers Listserv](#).

Questions and suggestions should be referred to Chris Allen at callen@calepa.ca.gov.

Table of Contents

Section 1: Introduction.....	2
CERS EDT in a Nutshell.....	2
Typical CERS Data Exchange Scenarios.....	3
Section 2: Recommended Approach for CERS EDT Technical Implementers	4
Section 3: Facility Submittal EDT Transactions.....	5
Section 4: CME EDT Transactions (Inspections/Violations/Enforcements).....	6
Appendix A: Sample .NET Code to Access Facility Submittal Export Endpoint	9

Section 1: Introduction

The [2009 legislation](#) requiring regulated business and Unified Program agencies to report program data electronically put in motion the development of the California Environmental Reporting System (CERS). In addition to providing a web-based user interface for businesses and regulators to access CERS, the legislation mandates electronic, machine-to-machine transfer of data using XML. Cal/EPA named these machine-to-machine transfers as CERS EDT (Electronic Data Transfer).

Implementing CERS EDT has proven to be a daunting task, both because of the legislative mandates and past agreements made by CUPA governance bodies, as well as the general complexity of the Unified Program--multiple interrelated forms, some with supplemental documentation, spanning multiple programs and managed by multiple regulators using sometimes differing business processes. In particular, the need to support facility submittal data originating from multiple locations (CERS web user interface, local regulator web portals, proxy submittals from local data systems, direct submissions by multi-facility businesses) is a far more complex model than the typical one-way transmission of data from a lower ranking government entity to a higher one (e.g., local→state, state→federal).

Thus CERS EDT implementation is quite complex and implementers will need a strong technical background and should anticipate a non-trivial investment in time to implement the services. This manual, along with the [CERS EDT Endpoint Documentation](#), provides the specific details on the “how” of using CERS EDT services—it generally skips explaining the “why” of design approaches. For additional information, please review the CERS EDT Home Page at <http://cers.calepa.ca.gov/EDT/> or contact Cal/EPA’s Technology Services Unit (TSU) staff. The [CERS2 Business Rules](#) document provides the details of various business rules that EDT implementers would need to consider when performing submit transactions to CERS.

CERS EDT in a Nutshell

CERS users (CUPAs/PAs) can gain authorization for production use of CERS EDT by agreeing to the [CERS EDT Data Exchange Agreement](#), establishing a CERS-Key account dedicated to data exchanges for that regulator/organization, and demonstrating successful test exchanges to Cal/EPA in advance of being authorized for production usage. Authorized EDT users communicate to SSL-secured, REST-based endpoints using HTTP messages consisting of an authentication header, and 1) for query-related endpoints provide certain URL and/or querystring values, and/or 2) for data submission endpoints, provide specified XML data or a ZIP document meeting specific requirements. ZIP document-based submissions accommodate sending document attachments or submissions with an extensive amount of data. All EDT data submissions to CERS include an XML document with data rigorously formatted using Cal/EPA-provided XML schemas.

If a **single** record in a CERS data submission generates any XML schema or business rule errors, the **entire** data submission (all records) is rejected. All CERS data submission endpoints return a schema-defined XML response about the success/status of the transaction and detailed error/advisory messages.

In general, the CERS user interface and EDT services use the same submittal and validation logic for consistent processing and error reporting. The submission EDT XML schemas provided by Cal/EPA express as many of the CERS data submission business rules as possible using XML schema (XSD) markup (i.e., hierarchical entities, data types, enumerations, choice sequences, and restriction patterns). Most of these business rules involve general

data element validation rules related to type, length, and allowable code values, which are adopted as state regulations and expressed in the [Unified Program Data Dictionary](#). Other business rules involving multiple fields on the same form and/or field dependencies across forms are documented in the [CERS2 Business Rules](#) documentation. Some business rules specific to EDT are discussed in the [EDT Endpoint Documentation](#). EDT implementers must apply this same business logic in their own systems if they intend to support data entry of new facility/CME submittals for eventual EDT submittal to CERS.

All EDT submittals must be performed by the CUPA on behalf of any PAs they may have. PAs (and CUPAs) can gain authorization to export/download XML submittal data as needed/desired.

Most submission endpoints will require regulators to provide unique local identifiers for their data entities (e.g. UST Tank, Inspection, etc.). The relevant web service API will return CERS unique identifiers and any other return values for the submitted data entities.

Although Cal/EPA originally envisioned businesses having the ability to directly submit their facility submittal data to CERS in XML format, a low level of interest from the business community resulted in these EDT exchanges being deferred for several years and when more definite interest and intent is expressed by regulated businesses.

Typical CERS Data Exchange Scenarios

CERS was designed so California's 100+ CUPAs/PAs could interact with CERS and CERS EDT in several different ways to meet their local practices/needs. Listed below are the three primary UPA/CERS usage scenarios.

Scenario	Description	CERS EDT Endpoints
CERS-Only	<ul style="list-style-type: none"> All regulated facilities are directed by their CUPA (and PAs) to always use the CERS Business Portal for their facility reporting. Regulators use the CERS web portal to review and act upon facility submittals, as well as manually enter (or upload via a specific MS Excel format) their CME data (inspections/violations/enforcements). Authorized CUPAs/PAs could choose to export/download data using the Regulator Facility Submittal Query and/or CME Query endpoints. 	None
CERS + Local System	<ul style="list-style-type: none"> All regulated facilities are directed by their CUPA (and PAs) to use Cal/EPA's CERS web portal for their facility reporting. The regulator's local system queries and downloads facility submittal XML data from CERS into the regulator's local system via EDT. Regulator staff review and act upon the submittals in their local system. The local system submits review/status data via Regulator Facility Submittal Action Notification EDT transactions. New facilities not in CERS must first be established via Facility Create endpoint. Regulators can send corrected Facility data via Proxy submittals to the Facility Submittal Submit endpoint. Regulator staff enters their CME data (CME) into their local systems, and then the local system submits new and updated CME XML data to the CME Submit endpoint*. 	<ul style="list-style-type: none"> ➤ Regulator Facility Submittal Query (RFSQ) ➤ Regulator Facility Submittal Submit (RFS) ➤ Regulator Facility Submittal Action Notification (RFSAN) ➤ Regulator Facility Create (RFC) ➤ CME Query (RCMEQ) ➤ CME Submit (RCMESS)
Local System to CERS	<ul style="list-style-type: none"> Regulated facilities are encouraged by their CUPA (and PAs) to use the CUPA's local reporting portal for their facility reporting because the portal provides added value beyond the features provided in CERS (e.g., permitting, billing). However, businesses can choose whether they want to use CERS or the local reporting portal (i.e., multi-jurisdictional businesses likely will want the 	<ul style="list-style-type: none"> ➤ Regulator Facility Submittal Query (RFSQ) ➤ Regulator Facility Submittal Submit (RFS) ➤ Regulator Facility

	<p>centralized reporting features in CERS).</p> <ul style="list-style-type: none"> • The regulator’s local system queries and downloads facility submittal XML data originating in CERS into the regulator’s local system via EDT. • Regulator staff reviews and acts upon submittals in their local system(s)**. • The local system submits facility submittals originating from the regulator’s local reporting portal (or proxy submittals from their local system) to the Regulator Facility Submittal Submit endpoint. New facilities not in CERS must first be established via Facility Create endpoint. The local system also submits Regulator Facility Submittal Action Notification XML transactions for CERS-generated Facility Submittals and/or any Facility Submittals the regulator is not immediately acting upon (status <i>Submitted</i> or <i>Under Review</i>). • Regulator CME data (CME) into their local systems, and then the local system submits new and updated CME XML data to the CME Submit endpoint*. 	<p>Submittal Action Notification (RFSAN)</p> <ul style="list-style-type: none"> ➤ Regulator Facility Create (RFC) ➤ CME Query (RCMEQ) ➤ CME Submit (RCMESS)
--	--	--

*A regulator could also opt to ONLY use the CERS web portal for managing their CME Data. CME Data can be entered manually in the CERS user interface, or uploaded in “flat file” format using a rigorously-defined, Microsoft Excel spreadsheet template.

**A regulator’s local system(s) may involve only a local web portal, or a local web portal communicating with some other local system (only one of these systems can communicate with CERS).

Section 2: Recommended Approach for CERS EDT Technical Implementers

Step 1) Get a CUPA Sponsor

EDT implementers who provide services to CUPAs (vendors) will need a CUPA to sponsor them in testing EDT services, presumably because they are purchasing data services/software from the vendor. Implementers directly employed by their CUPA should have their management contact Cal/EPA indicating the CUPA is intending to implement EDT services with their own staff resources.

Step 2) Review all EDT Technical documentation listed on the CERS EDT Home Page

(<http://cers.calepa.ca.gov/EDT/>). The key documents are this EDT Manual, the [CERS2 Business Rules](#), the [EDT Endpoint Documentation](#), and the [CERS Data Registry](#).

Step 3) Subscribe to Cal/EPA’s CERS2 EDT Listserv

The CERS Electronic Data Transfer Implementation listserv is a free service; you will receive periodic e-mails with the newest CERS EDT Implementation listings. Subscribe at <http://www.calepa.ca.gov/Listservs/ListSubscribe.asp?LID=110>.

Step 4) Study CERS Data Registry/Data Dictionaries

Become familiar with the Unified Program Data Dictionary by using the CERS Data Registry

- [Search/query](#) or [download](#) the Unified Program Data Dictionary Version 3.10
- [Search/query](#) or [download](#) the CERS System Fields Data Dictionary. CERS System Fields are data elements not specifically related to facility/CME reporting, but are required for implementing the CERS web portal and CERS EDT services.
- [Search/query](#) or [download](#) the CERS Supplemental Fields Data Dictionary. CERS Supplemental Fields are data elements that apply to facility reporting/management, but are not adopted into regulations and thus are optional.
- If you have data which may map to the past/original Title 27 Unified Program Data Dictionary definitions, review [“Summary of Data Exchange-Related Changes to the Unified Program Data Dictionary”](#)

Step 5) Map Local System Fields to CERS Data Dictionary Fields

This may require extensive assistance from program and other knowledgeable staff about the local data systems. The staff performing this mapping must identify where local fields diverge from Unified Program Data Dictionary data elements so programmers can develop the transformations and/or interpolations necessary to populate CERS submittal data.

Step 6) Develop Code to Create XML Submittal Files

If you are only using query endpoints (Tier 2), you can skip this step. But if you are using any endpoints that submit data into CERS, you will need to develop code to transform mapped local fields into XML packages following the appropriate CERS XML data exchange schemas. The majority of CERS business rules are expressed in the schema validations, so developers can and should extensively test and analyze their draft XML submittals against the published schemas before they attempt test submittals against CERS EDT Services.

Step 7) CUPA Establishes a Data Exchange Agreement with Cal/EPA

In order to test EDT endpoints, the CUPA will need to send an email to Cal/EPA approving the EDT implementer to use the testing services and to view a copy of their production facility data. If an UPA is working with multiple vendors, they will need to complete an agreement related to each vendor. See the [CERS EDT Home Page](#) for more information.

Step 8) Establish a Dedicated CERS Account for EDT Transactions

To use the CERS EDT Services, each UPA will need to establish a CERS-Key account dedicated exclusively for EDT transactions. If the UPA is working with multiple vendors, a different dedicated EDT Account will need to be established and configured separate for each vendor. TSU staff will ensure this account has rights to access the REST endpoints on behalf of the UPA (and its PAs where relevant). This account can ONLY be used for EDT Access, so it can't be an existing user's account. The UPA needs to [create the CERS Account](#) in advance of sending the EDT Access request email. Since each CERS Account requires a unique email address, the UPA will need a unique email address to create its EDT Access Account.

Step 9) Performing Test Submittals

Once an exchange agreement has been received and the dedicated CERS-Key account established, TSU staff will establish rights for the UPA/vendor to initially access a test version of the CERS EDT services. Once the regulator (and/or their vendor/contractor) has demonstrated an understanding of EDT processes and has successfully used all relevant test services, TSU staff will grant additional rights to the dedicated CERS-Key account to perform production submittals.

There are a number of tools Cal/EPA provides EDT implementers for EDT testing. These include an open source Windows EDT client you can use to invoke EDT endpoints. And in the Regulator Portal, a CERS EDT dashboard is provided for transaction review, EDT account authorization, and other EDT-related activities. The EDT dashboard feature is also available in the Production version of CERS for implementers to monitor production activity.

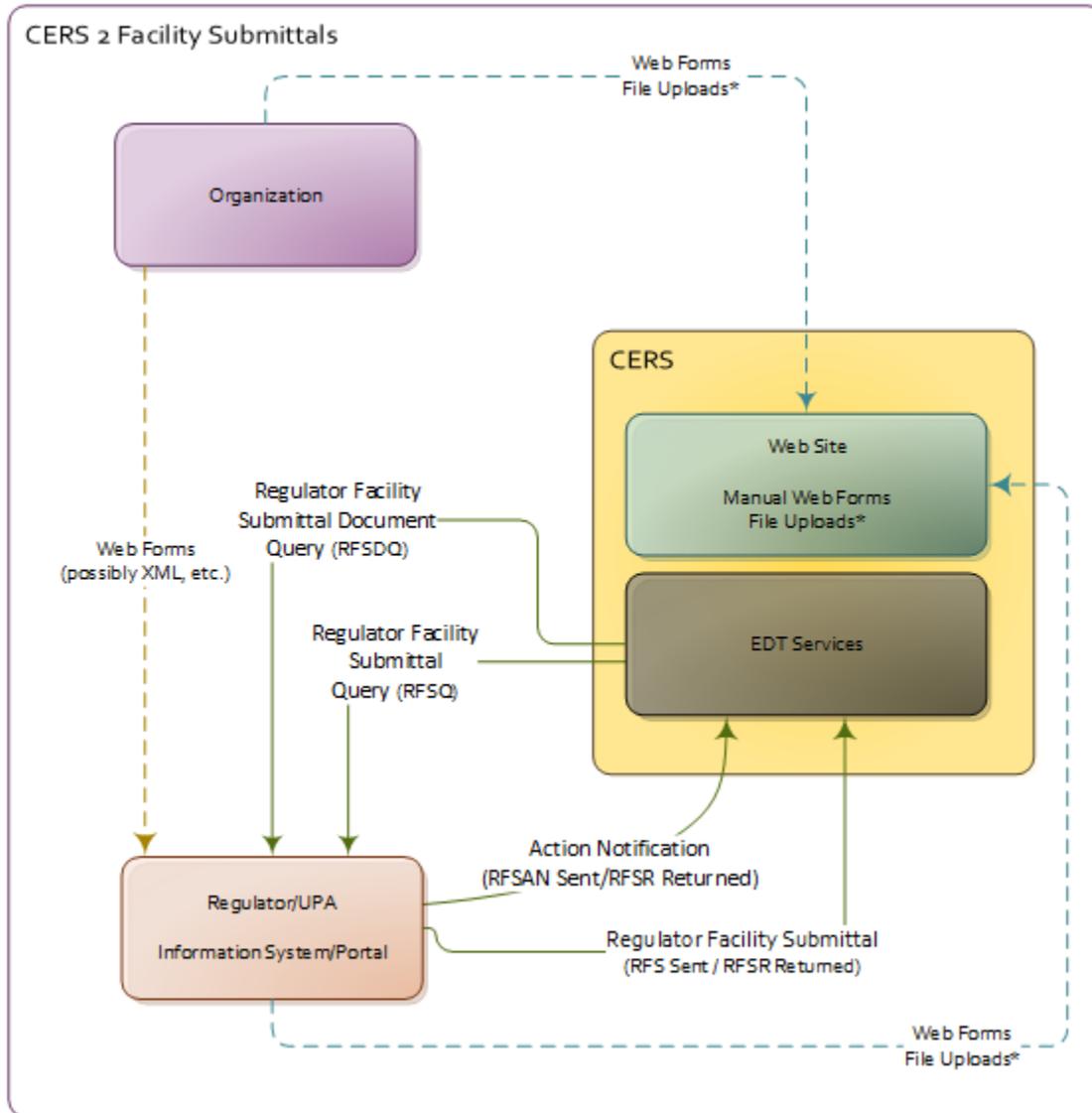
Section 3: EDT Endpoints

The [EDT Endpoint Documentation](#) lists all of the CERS EDT endpoints. The endpoints are grouped into numbered "tiers" representing increasing business process complexity and security considerations. UPAs will need to

request access from Cal/EPA for the specific tiers they wish to use. Information about UPA EDT data exchange agreements for use of EDT Services is available at <http://cers.calepa.ca.gov/EDT/ExchangeAgreement/>.

EDT implementations involving only extracting data out of CERS only need access to query-related Tier 1 and Tier 2 endpoints. The remaining EDT tiers involve submitting various types of data to CERS—not every implementation will need all of these tiers.

Shown below are the primary facility submittal-related EDT data flows and their respective endpoints.

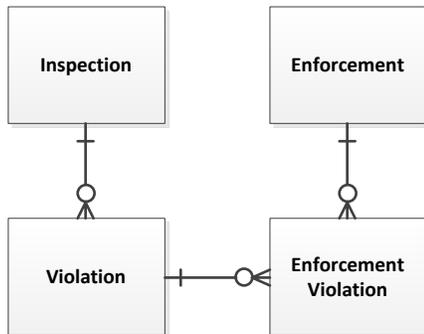


*Flat file uploads (MS Excel, CSV) supported for Inventory and CME data only.

Section 4: CME EDT Transactions (Inspections/Violations/Enforcements)

This section discusses general business rules, data flows, system fields, and conceptual XML schemas structures needed for electronic data transfer of Inspection, Violation, and Enforcement data. Most CERS documentation will use the acronym “CME” to generically refer to this data—CME is used in US EPA’s RCRA-related data systems and stands for “Compliance, Monitoring, and Enforcement.”

CME EDT is a **one-way** flow from regulators to CERS (Cal/EPA). One of the key differences between CME and Facility Submittals EDT is update and deletion requests can be transmitted for CME entity records previously submitted by the regulator (e.g., to provide a Return-To-Compliance Date). There are four CME data entities involved in CERS CME.

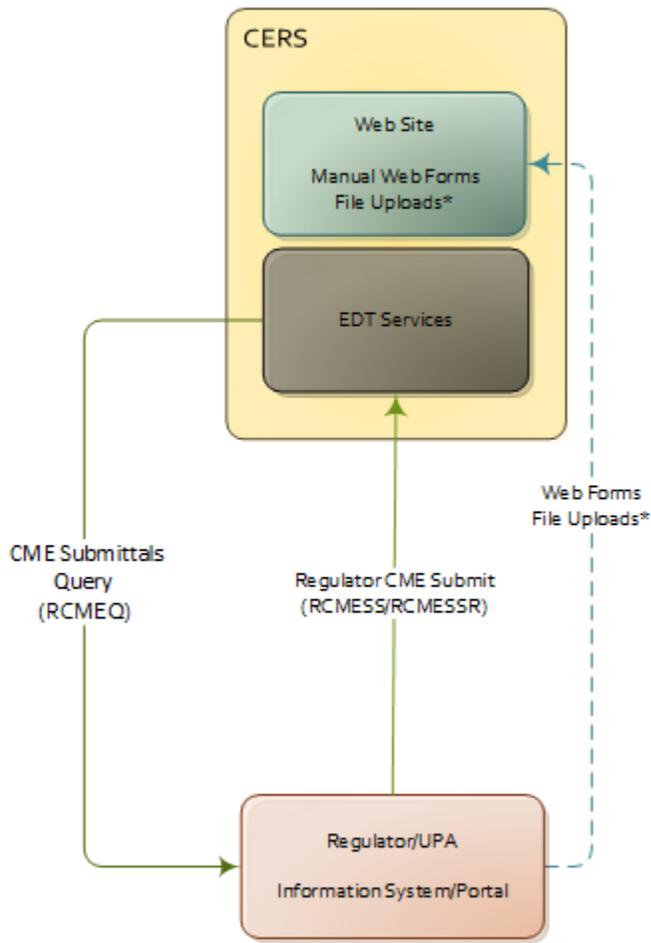


Although some earlier versions of the Unified Program Data Dictionary and XML Schemas allows submittal of only summary violation information, beginning in January 2014 regulators must provide violation detail data for all inspections (not just the Large Quantity Generator violations required to meet California’s reporting requirements to US EPA).

Like all EDT-based submitted data, only CUPAs can submit CME data via EDT, so CUPAs with PAs would generally have responsibility for collecting/compiling CME data from PAs. PAs can also manually enter CME data in the CERS UI for the submittal elements/facilities they are specifically responsible for.

Although not part of EDT, the CERS user interface will support “flat file” uploading of CME data using a rigorously defined Microsoft Excel spreadsheet workbook template that includes 4 worksheets (one for each CME data entity). Essentially, the entire workbook will be transformed into a CMESubmittal.xml file and sent through the EDT submittal logic.

CERS 2 Inspections/Enforcements (CME)



*Flat file uploads (MS Excel, CSV) supported for Inventory and CME data only.

Appendix A: Sample .NET Code to Access Facility Submittal Export Endpoint

```
using System;
using System.Linq;
using System.Text;
using System.Net;
using System.IO;
using System.Xml.Linq;
using System.Security.Cryptography;

namespace CERS.EDTServices.Client
{
    public class FacilitySubmittalImportClient
    {
        public void Submit()
        {
            //Define username & password variables; substitute with valid CERS-Key username & password
            string userName = "[AuthorizedCERSUserName]";
            string password = "[CERSPassword]";

            //Use SHA1 to generate a hash string to put in HTTP Authorization header below.
            SHA1Managed sha1 = new SHA1Managed();

            //Generate hash bytes.
            byte[] hashedPassword = sha1.ComputeHash(Encoding.UTF8.GetBytes(password));

            //Convert the hash bytes to a Base64 string to be sent to the service.
            string hashedPasswordString = Convert.ToBase64String(hashedPassword);

            //Setup and make Web Request.
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://.../Regulator/98761/FacilitySubmittal/");
            request.Method = "PUT";
            request.ContentType = "application/zip";

            //Add Http Authorization header
            request.Headers.Add("Authorization", string.Format("user {0}:{1}", userName, hashedPasswordString));

            //Fill the byte array with your data
            byte[] byteStream = new byte[100];

            //Write data to stream
            request.ContentLength = byteStream.Length;
            using (Stream writer = request.GetRequestStream())
            {
                writer.Write(byteStream, 0, (int)request.ContentLength);
                writer.Flush();
            }

            //Get the response from the service.
            HttpResponseMessage httpResponse = (HttpResponseMessage)request.GetResponse();
            if (httpResponse.StatusCode != HttpStatusCode.OK)
            {
                //Get the FacilitySubmittalResponse XML data back.
                XElement xmlResponse = XElement.Load(httpResponse.GetResponseStream());
                //Parse xmlResponse and check for status.
            }
        }
    }
}
```