

Automated Logic Corp.'s OpenADR Add-On

What is OpenADR?

OpenADR (Automated Demand Response) is a high-level computer-to-computer communication protocol, a product of the Smart Grid initiative. Its purpose is to provide a standard way for utility suppliers to automatically and securely communicate with their customers' building systems to reduce energy demand during critical periods. OpenADR can also be used to encourage voluntary reductions and rescheduling through dynamic (real-time or forecast) time-of-day pricing. An early version (OpenADR 1.0) was primarily used as an experimental development platform by a few utilities in North America. The current version (OpenADR 2.0) was released in late 2013 and is being adopted and gaining recognition as an international standard. Implementation of version 2.0 is still in the early stages and as with any new standard, some issues and conflicting interpretations are being resolved through experience.

Why would a customer want to implement OpenADR? The primary reason is to reduce utility costs. Utility companies have to design and operate their generating plants and transmission lines to meet the highest possible demand. If they can reduce peak demand they can save money. Therefore, they are willing to incentivize their customers to reduce energy consumption at peak hours by offering reduced rates and other incentives for participation in automated demand response. The OpenADR protocol does not concern itself with these incentives, nor does it have anything to say about how the customer will reduce demand. The protocol simply provides a way for the utility supplier's computer to communicate with the customer to initiate these reduction efforts. The protocol is flexible enough to cover a wide range of systems, from a simple on/off rooftop air conditioning unit to a college campus building automation system that can fine-tune thousands of pieces of equipment as well as switch to alternative energy sources. How much demand the customer can and will shed, and how much of an incentive the utility supplier will give them for doing so is a matter to be worked out between the supplier and the customer. Similarly, how the customer is going to reduce demand to fulfill the agreement is not part of the protocol. The protocol simply provides a means for the utility supplier to tell the customer when and how much demand reduction is needed.

What's the difference between OpenADR Part A and OpenADR Part B?

The OpenADR standard includes requirements for "OpenADR 2.0a"¹ and "OpenADR 2.0b."² Products can be certified for compliance with either, or both, sets of requirements. 2.0a is a simpler set of requirements, essentially a subset of the 2.0b requirements. OpenADR 2.0a provides a method for a utility provider to use a secure HTTP connection to inform the customer of upcoming Demand Reduction

events. The events are limited to a signal type called a “SIMPLE” signal because it is a simple sequence of times when the demand reduction level (or price) will be normal, moderate, high or special. For this to be useful, the utility company and the customer must have previously come to an agreement as to what these different levels mean and what response actions the customer will take at each level. (The term “special” is simply a name assigned to a demand reduction level. The utility provider and the customer need to define what response action should be taken for a “special” event.)

OpenADR 2.0b includes a more complex set of requirements. More options for transport protocols and security levels are available, and a much wider variety of events can be forecast.

To understand the different types of events, it is necessary to have a better understanding of what constitutes an “event.” A typical event is shown in Figure 1:

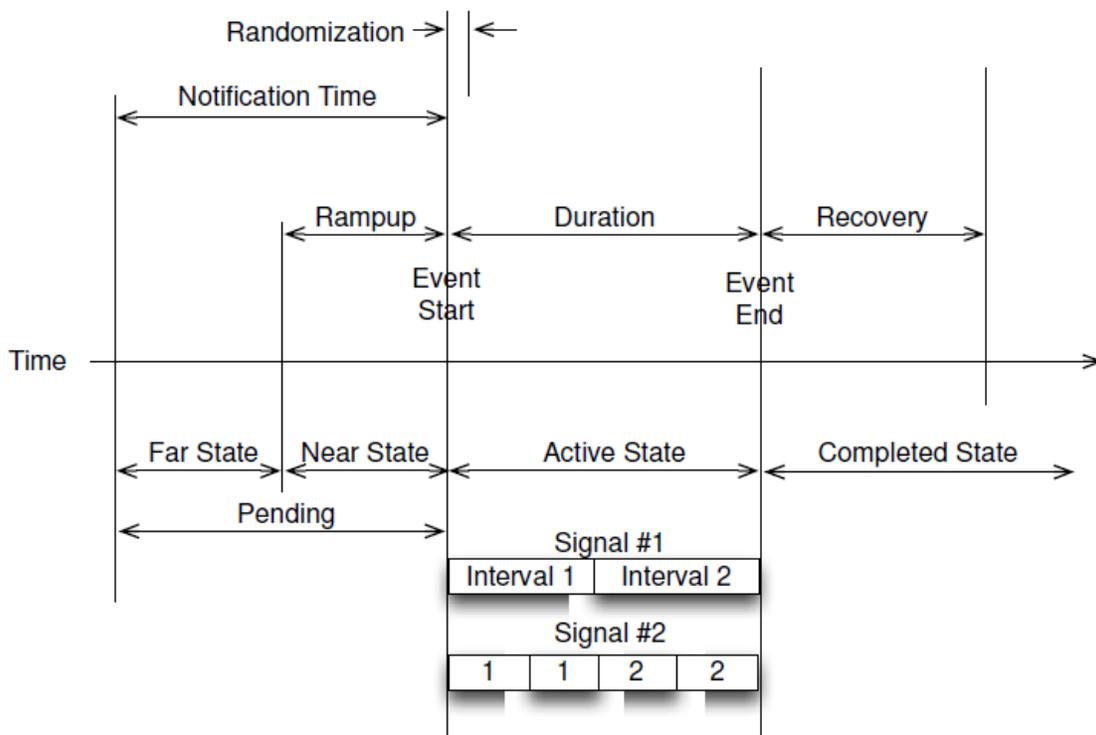


Figure 1: A Typical OpenADR Event

(Courtesy: ©OpenADR Alliance 2013)

Note that the utility provider may specify a ramp-up time before the event actually begins, to give the customer time to gradually change their energy use and prevent any sudden disruption of operations. Similarly there may be a specified recovery time after the event, during which the customer would gradually return to normal operations. The “randomization” factor staggers the beginning and end time of the event so large pieces of equipment are not all turning off at the same time when the event begins, and more importantly, aren’t all turning on at the same time when the event ends. The actual Demand Response Event is confined to the time period shown as the “active state.” This example shows two signals occurring during the event. Signal number one may correspond to different utility

prices during interval one and interval two, while signal number two might correspond to different demand reduction levels. OpenADR 2.0b allows multiple signals during an event. OpenADR 2.0a allows only one signal during an event, and this signal is limited to the normal, moderate, high, or special states mentioned previously. If signal number two in the example above were of the “SIMPLE” type allowed by 2.0a, the enumerated values of “1 – 1 – 2 – 2” would correspond to “moderate – moderate – high – high.”

What types of signals can be communicated using 2.0b? In addition to the “SIMPLE” enumerated values allowed under part 2.0a, 2.0b allows the utility supplier to specify an energy price for a given interval, either in absolute terms, as a change to the existing price, or as a multiplier to the existing price. The utility supplier can also specify a maximum load, either in absolute terms, as an offset to some agreed-upon baseline (which could be the current load), as a multiplier of some agreed-upon baseline, or as a discrete level. There are also several options to specify load control based on capacity or setpoints of individual pieces of equipment, such as duty cycling a fan or adjusting the setpoint of a rooftop unit. For specifics, see the OpenADR 2.0b Profile Specification.³ (Please note that the initial release of ALC’s add-on does not support all of the signal types defined by the OpenADR specification. If there is a need for a signal type that is not currently supported, please contact us.)

What if a customer doesn’t want to participate in an event?

It’s possible that a utility provider may call for a demand reduction event while a customer is in the middle of a critical operation. The OpenADR protocol does provide methods for a customer to opt out of a particular event, or even to tell the provider in advance that they cannot participate in demand reduction events during certain time periods. Like all implementation details, this is subject to whatever arrangements the provider and the customer have made outside of the protocol. There may be a financial penalty for opting out of an event, or for opting out of more than a certain percentage of events. It’s possible the utility provider may not want to allow the customer to opt out of any events. The procedure for providing a schedule of times when a resource is or is not available to participate in demand reduction events is specified in the OpenADR 2.0b specification and is supported by ALC’s OpenADR add-on. However, that procedure is described as being “for future implementation,” so some utility providers may not currently support it. The OpenADR 2.0a specification does not provide a way to schedule availability in advance, but both OpenADR 2.0a and 2.0b allow customers to opt out of a scheduled event. The ALC add-on lets customers see what events are scheduled, and the add-on’s user interface provides a manual command to opt out of any scheduled event.

How can a customer implement OpenADR?

First, the customer should meet with the utility provider. OpenADR 2.0 is a new protocol and not all providers support it. If it is supported, the customer and provider will need to come to an agreement as to what resources are available to participate in demand reduction events, how much demand the customer can shed upon request, and what types of demand reduction events the customer’s system can respond to. At some point, it may be advantageous to have a representative from the local ALC field

office participate in these negotiations, to make certain the ALC OpenADR add-on is compatible with the events and reports the utility provider will be using.

In parallel, the customer should analyze their system to determine what types of demand reduction actions are acceptable. The customer will want to know what types of demand reduction programs are available before spending the resources to determine a demand reduction strategy, but the customer will not be able to conclude an agreement with the utility supplier until they are ready to commit to specific demand reduction actions. Depending on the customer's operations, there may be several demand-reduction actions that can be implemented for at least a short time without unduly affecting comfort and productivity. Relaxing zone setpoints is an easy way to reduce demand in an ALC system, as there are four levels of setpoint relaxation built into the ALC setpoint microblock. This has also been shown to be the most effective way to reduce demand without disturbing occupants.⁴ Additional savings may be possible by adjusting temperature and variable frequency drive setpoints on air handling units, chilled water systems, and boiler systems. The customer may have decorative lighting, vehicle charging stations, and other non-critical systems that can be turned off completely during a demand reduction event. The actions which a customer is willing to take to temporarily reduce demand depend on the facilities and operations the customer controls. It may be necessary to test these actions while monitoring the customer's energy use to determine how much energy each action can save. The customer will need to prioritize these actions so they can be activated in an incremental fashion, either to implement pre-determined demand reduction levels (as in the SIMPLE Signal of OpenADR 2.0a) or managed by a closed loop control algorithm to meet specific energy reduction targets as required by some of the more complex load control actions specified by OpenADR 2.0b.

Once the response actions are prioritized, they must be expressed as a sequence of operations and programmed into WebCTRL[®] control logic. This could be done by a knowledgeable customer, but in most cases will be done by the ALC field office that supports the customer's site. The ALC OpenADR add-on comes with LogicSymbol files that will receive each type of demand reduction event signal supported by the ALC add-on. A screen capture of the logic for a SIMPLE demand reduction event is provided as an attachment. The ALC OpenADR add-on will write information to parameters in this logic, and the logic will convert this raw signal data into information which will be needed by WebCTRL to implement the required demand reduction. For the SIMPLE event logic illustrated in the attachment, this information includes:

- The current demand level (0, 1, 2, or 3)
- The number of minutes until the next occurrence of Demand Level 1
- The number of minutes until the next occurrence of Demand Level 2
- The number of minutes until the next occurrence of Demand Level 3

This information is exposed to the WebCTRL system as BACnet[®] Status microblocks. The customer or an ALC field office programmer will need to create custom logic which reads this information and executes the sequence of operations developed for the customer's site.

Developing the sequence of operations and the custom logic to implement it is probably the most difficult step to implement OpenADR. Once this is done, the customer can purchase the ALC OpenADR add-on through their local field office. It will be available to the field office on the ALC website, but it is a licensed add-on and will not work until the field office purchases a license through ALC customer service. The field office and the customer can then install it on the customer's WebCTRL server and register this add-on with the utility provider so they can begin receiving demand reduction events.

Is ALC's OpenADR add-on certified?

ALC's OpenADR add-on is not currently certified. ALC is a member of the OpenADR Alliance and it is our intention to pursue certification of our interfaces to the OpenADR 2.0a and 2.0b protocols. At present, these are new protocols, not yet implemented by many utility suppliers. Because ALC intends to update its interfaces whenever a new issue or a new interpretation requires a change, ALC has decided to delay certification until the standard is more mature.

References:

1. OpenADR 2.0 Profile Specification: A Profile, OpenADR Alliance, <http://www.openadr.org/specification>
2. OpenADR 2.0 Profile Specification: B Profile, OpenADR Alliance, <http://www.openadr.org/specification>
3. Automated Logic Technical Documentation for the OpenADR Add-on
4. Findings from the 2004 Fully Automated Demand Response Tests in Large Facilities, <http://eetd.lbl.gov/sites/all/files/58178.pdf>

1 Attachment

Sample control program to receive a SIMPLE demand response signal

Attachment 1: Sample control program to receive a SIMPLE demand response signal:

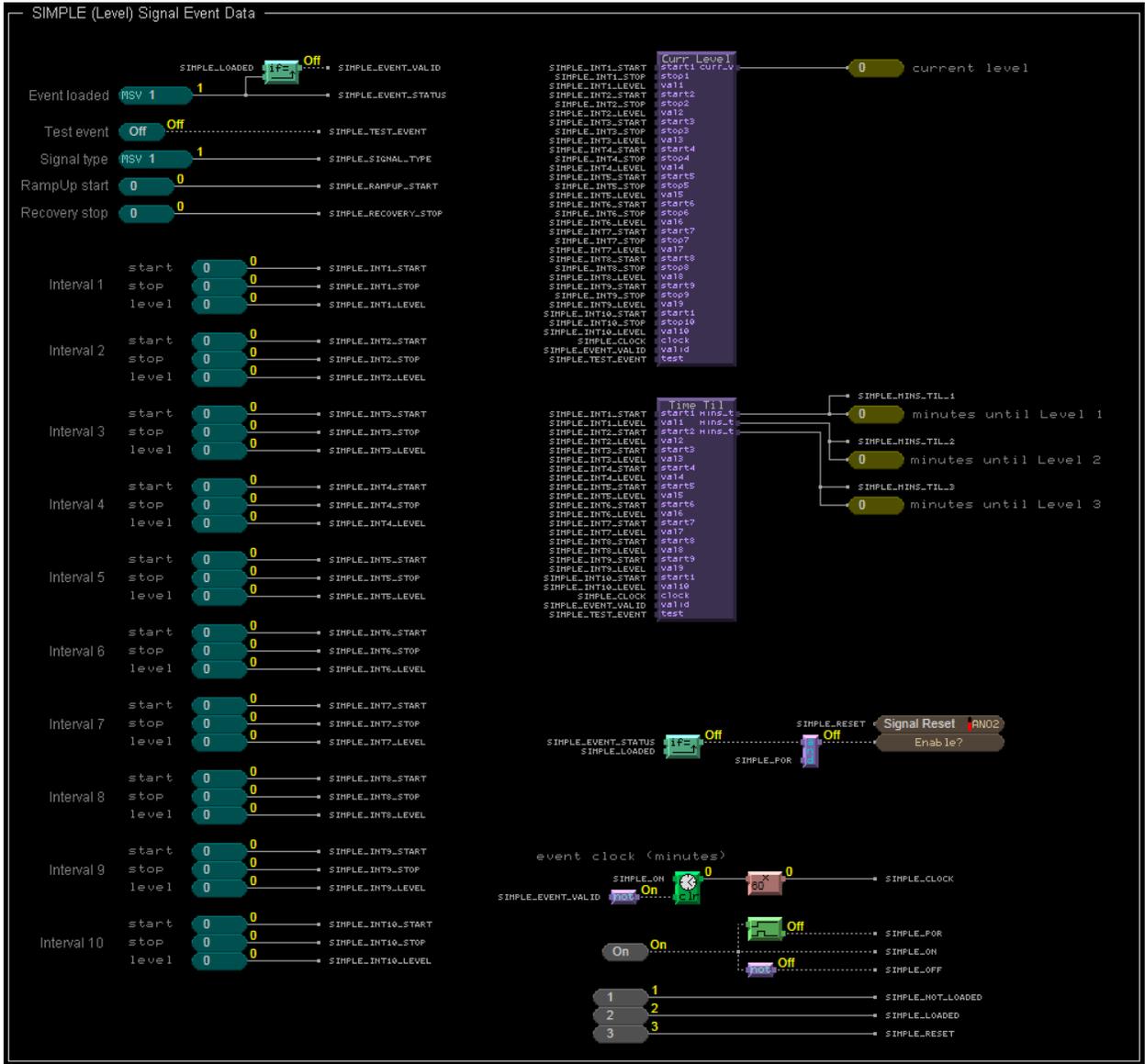


Figure 2: Control Program for Receiving SIMPLE Signal Type