Cedexis Radar

Configuration Guide

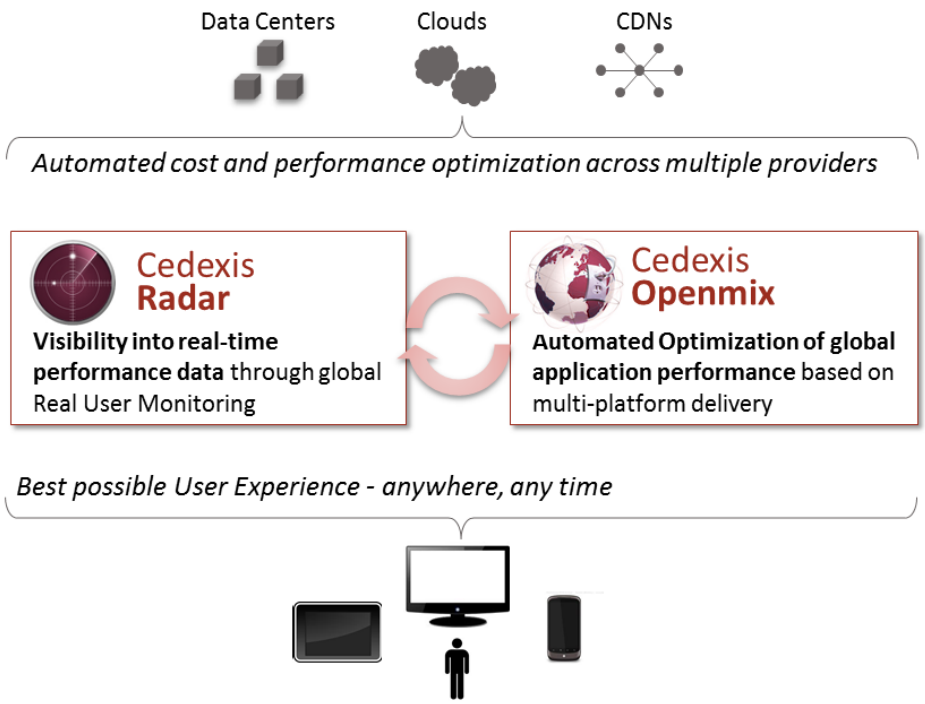
Contents

Introduction to Cedexis

Cedexis was started in 2009 by former Akamai executives, Marty Kagan and Julien Coulon to build tools for large websites to efficiently leverage mutli-vendor sourcing of data-centers, Clouds, CDN's, etc.

Cedexis has combined real-user performance monitoring (or RUM) and data-driven DNS based global load balancing into a unified service. The service is unique in the sense that it employs end user based probes collecting real-time information (clients make requests for test objects) and a programmable DNS decision engine (called OpenMix) that can make use of this data.

The OpenMix service is important since it allows high flexibility load-balancing mechanism's to be configured using Radar and other data (this could be data you provide). Configuration scripts can support such obvious methodologies as sending the user to the fastest provider, or as subtle and sophisticated as factoring in any thing you can collect or think of (Time of day, cost, contractual commitments, weighting servers, capacity, BTU power usage, Electricity cost etc.)



Cedexis Radar

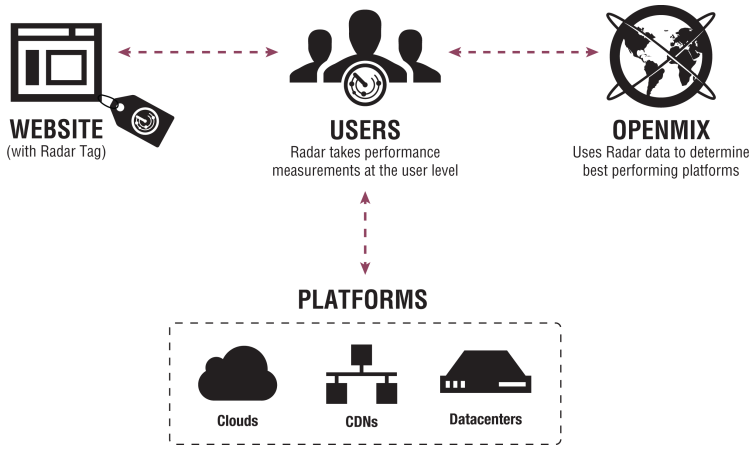
Radar is just one component part of Cedexis's offering. As a Real User Measurement (RUM) tool, Radar, provides transparency, validation and/or confirmation of your delivery strategy. In addition, Radar forms the backbone to the OpenMix service, where real-time performance/availability/through-put data can be used to load-balance between your infrastructure, cloud and CDN providers.

Cedexis Radar solves multiple problems with traditional methods of monitoring and data gathering. The key benefits to our customers and community with Radar are:

1. Massive testing environment, deployments in every network in every location (32,000+ recognized networks so far and growing).
2. Global view and not restricted by agent deployments, connectivity or costs (230 Countries).
3. Real data not gamed or influenced - Radar data is almost impossible to forge with real users doing the testing (1Billion measurements a day).
4. All users are not the same, different machines, connections, devices, get to the details.
5. Gain transparency into you provider and the platforms/clouds/delivery mechanism's you are not using.
6. Focus on the metrics that make a real difference to your users of you site/application (Performance, Availability, and Quality of Service).

Cedexis Radar is the Internet's first infrastructure monitoring cooperative and has placed reference objects on 25+ Cloud and CDN providers.

By adding a simple JavaScript tag to a page, Cedexis uses browser based measurements to download this reference objects and comparing internal & external infrastructure, multiple datacenters, delivery networks and cloud platforms as seen by ***your actual end-users.***



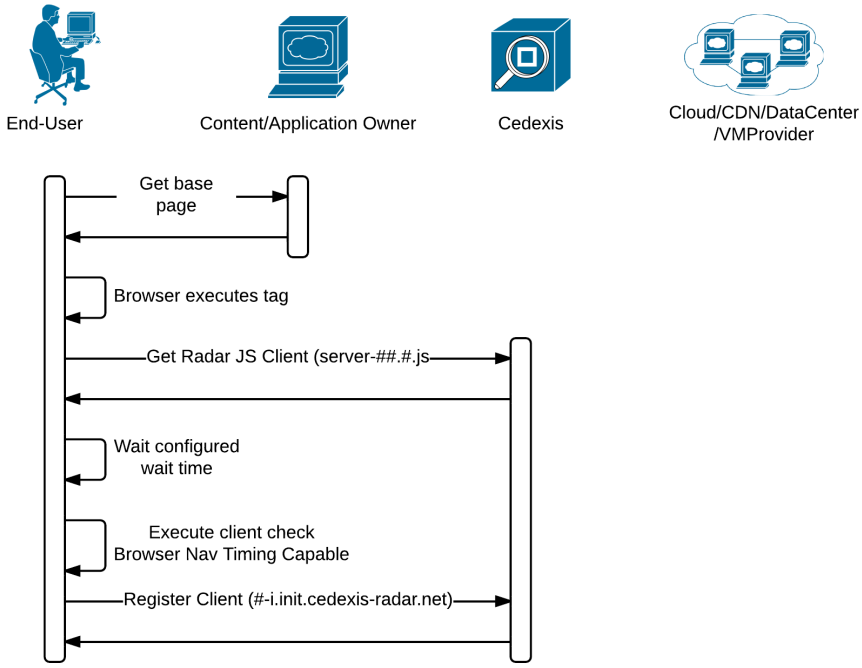
Cedexis Radar (http) – How it works in more detail.

Cedexis Radar forms the back-bone to the Cedexis data collection methodology. Radar uses a single JavaScript embedded within a content page or application provider's pages to feedback information on the performance and availability of a providers/owners data center or delivery strategy.

The JavaScript "Tag" has been specifically engineered **NOT** to interfere with the user experience of the page.

The recording and execution of the tag is broken into two main parts, the first is the initialization step, the second is the test sequencing dependent on the type of browser being used.

The initialization



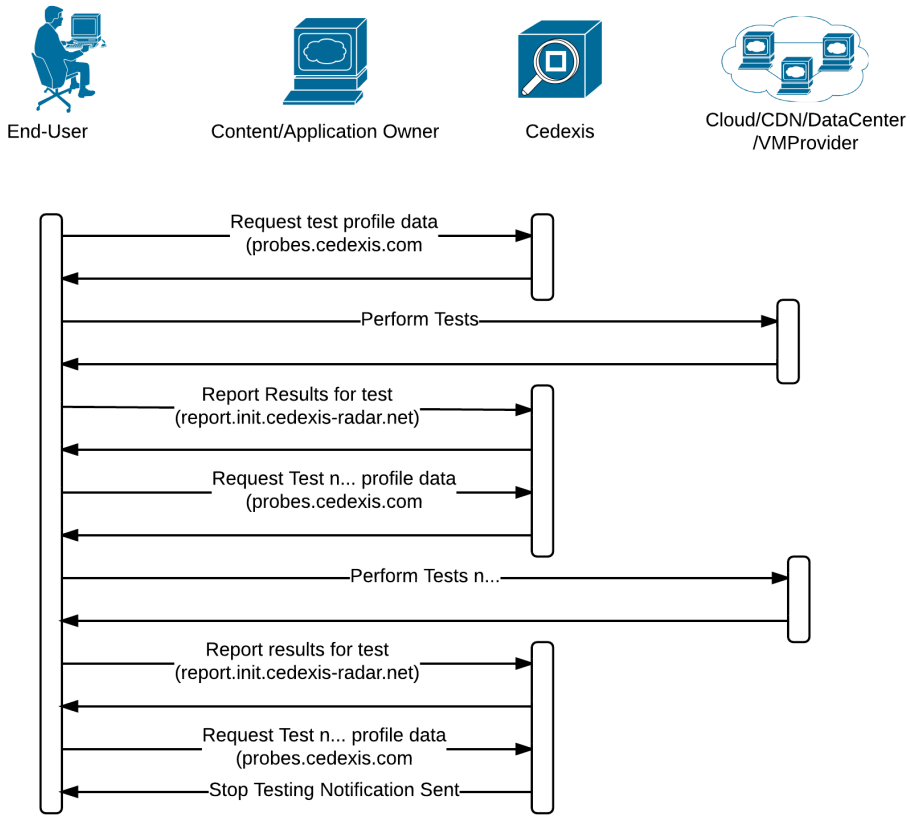
The following steps are completed by every end-users browser

1. The browser downloads the initial base page and executes the source code.
7. The JavaScipt is executed (most customer's choose to place the tag just before the closing </body> tag)
8. The Radar JS Client is requested from Cedexis.
9. The JavaScript waits for the page load to complete.
10. The JavaScript waits for a configured (in the portal) period of time – default is 2 seconds.
11. The JavaScript checks to see if the Browser is NavTiming capable.

12. The Radar JS client registers itself with Cedexis.

Test Sequencing – NavTiming Enabled

This form of testing is only completed on IE9+ on desktops and Firefox 8+ on desktops, all other browsers use the non-NavTiming method.



1. The Radar Client requests the first test profile from Cedexis, this contains information on which provider and what type of tests to make. When NavTiming is available, the test profile returned is to have the Radar Client report back on the Nav Timing elements on the Base page the JS was placed upon.

2. The Radar Client sends this data back to Cedexis.

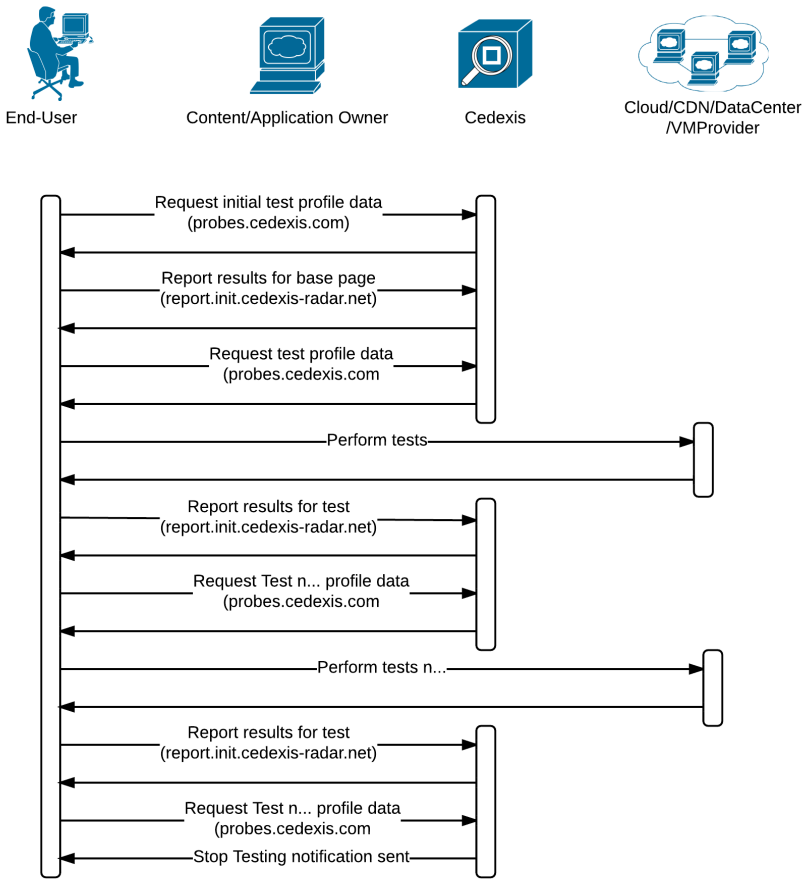
3. The Radar Client will then request another test profile. The profile typically contains the CDN/Cloud or Data Center to test and the type of tests to be made (e.g. Response Time, Throughput etc).
   The Radar client receives the test profile and makes a request for test-objects held either on CDN's Clouds or Data Centres (called "providers"). These tests are completed using a hidden iframe with performance information being recorded from the NavTiming array on the browser. Each provider test uses

two objects: a 1.1KB object called cdx.htm and a 102-KB object called cdxl.htm. The smaller object is requested twice, each time with a different random URI parameter, in order to prevent any intermediary caches from caching the content (effectively cache-busting downstream) and skewing any results.

4. Once the test is completed the Radar Client will report all the Nav Timing array components back to Cedexis.
5. The Client will continue fetching test profiles and executing tests until it is notified by Cedexis to halt.

Test Sequencing – Non-NavTiming Enabled

The non-NavTiming method uses a slightly different method in collecting the measurement statistics. This method is used for all browsers **that fall outside the one testing** above. For **clarity the non-**Navtiming method includes Chrome Browsers.



(Figure1)

1. After initialization the Radar Client requests a test profile from Cedexis.
2. The Radar Client then starts its test using two objects: a 50-byte object called cdx10b.js and a 100-Kbyte object called cds10b-100KB.js. The smaller object is requested twice, each time with a different random URI parameter, in order to prevent any intermediary caches from caching the content (effectively cache-busting downstream) and skewing any results.
3. The Radar Remote probing module takes the time measurements of the download and reports the results back to Cedexis.

The data generated by this process typically includes:

- Availability—whether the object loads or not.
- HTTP Connect Time—how long it takes for the browser to establish a connection with the server, inclusive of the DNS resolve time, the establishment of the initial TCP and HTTP sessions to the provider.
- HTTP Response Time—how long it takes for the server to respond to a subsequent request, once all of the noise of establishing a connection is completed. This is a relatively close approximation of TCP round-trip time (RTT) from the browser to the provider.
- Throughput—this is the data rate of the connection, in kilobits per second, as measured from the retrieval of the 100 Kbyte object.
- HTTP Page Load* – this is the time it takes for the browser to load the page and components including initial DOM delivery.
- Logical/Geographical location of user.

**Note: HTTP Page Load: this is only supported by those browsers that support the new NavTiming standard – e.g. IE9+, Chrome 6.5+, and Firefox 8+.*

Who is Cedexis Monitoring?

Cedexis deploys its standard test objects across multiple service providers in the CDN/Cloud Computing and Data Centre space. By using a consistent object a definitive view of performance and availability can be ascertained from the point of view of the end-user.

Examples of the providers Cedexis tests are as follows:

Akamai
Akamai (VIP)
BitGravity
Amazon EC2 Toyko
Amazon EC2 Singapore
Amazon EC2 Ireland
Amazon EC2 Virginia
Amazon EC2 California
Amazon S3 EU
Amazon S3 US
CacheFly
CDNetworks
CDNVideo
ChinaCache
ChinaNetCenter
CloudFlare
Cloudfront
Cotendo
Edgecast
Fastly
GoGrid
Google App Engine
Highwinds
Internap
Internode
Joyent
Level3
Limelight
NetDNA
Ngenix
Nirvanix
Rackspace CloudServer
Prime Networks
Voxel
Voxel VoxCloud Amsterdam
Voxel VoxCloud New York
Voxel VoxCloud Singapore
Windows Azure
Windows Azure Anywhere US
Windows Azure US North Central
Windows Azure US South Central
Windows Azure East Asia
Windows Azure Southeast Asia
Windows Azure North Europe
Windows Azure West Europe
Yacast

For more information, please see our measurements page:
http://www.cedexis.com/products/measurements.html

Integrating Cedexis tags

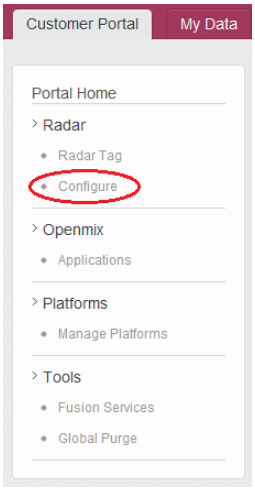
Integrating the Radar tag is relatively simple with the following steps to be completed.

1. Add this JavaScript snippet to the sites markup, preferably before but at the end of the <body>.

<Note this has not been included here as is specific to each customer>

Turning off Testing in an Emergency

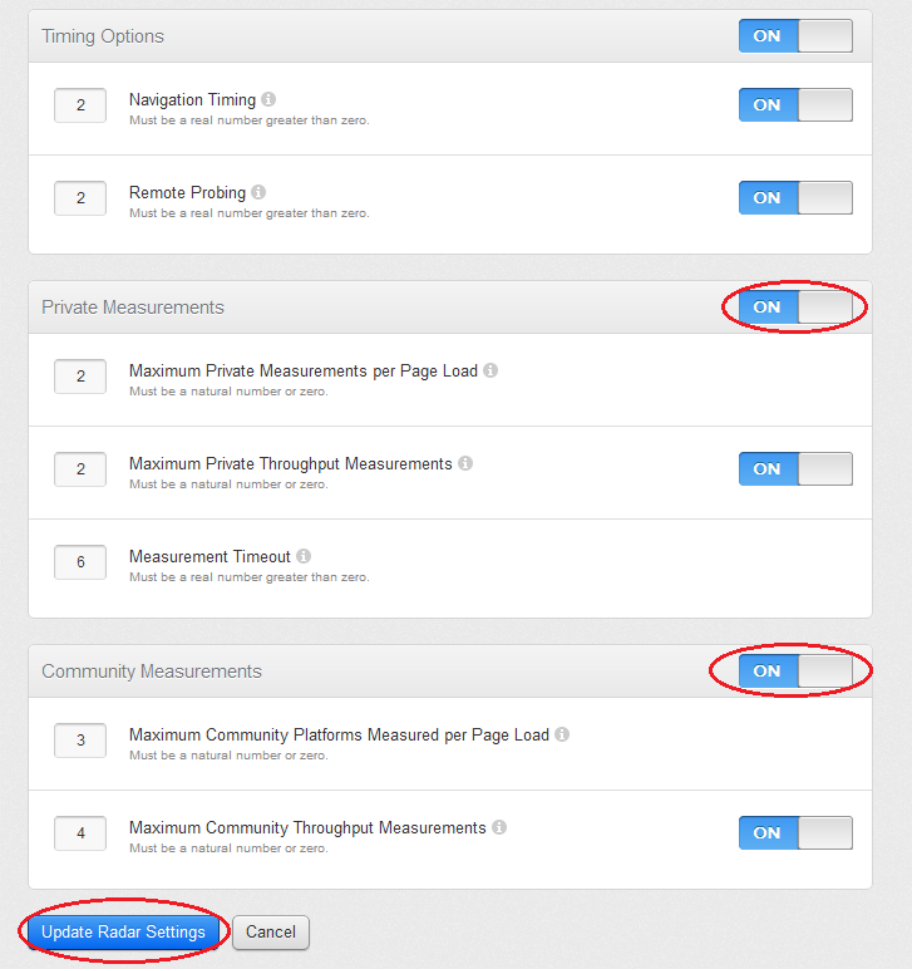
There may be a requirement to turn off testing of either a CDN or your Data Centre quickly. Cedexis has placed provisions for this within the Portal that allows the customer to stop testing immediately avoiding emergency code changes to the site, should something un-expected occur.



Configuration of the Radar testing service is completed from within the portal, ***under the "Customer Portal" Tab, under Radar "Configure".***

To Turn off testing of either Community Measurements, Private Measurements or both, select the ***ON/OFF switches, moving them to the OFF position.***

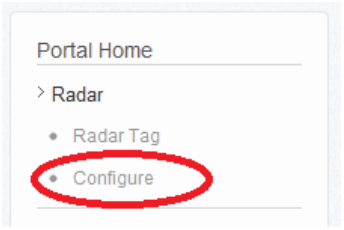
Finally ***click the "Update Radar Settings"*** to confirm the changes. Once completed this will cause the Radar JS to stop testing (this will usually occur within 1 minute).



Radar Tag Configuration items

The JavaScript has eight (8) parameters that can be used by the site/application owner to customize the Radar tag. These adjust timing and delay elements, number of tests completed by end-users for community and private measurements and time out values to measure availability.

To configure these options, log into your account at www.cedexis.com and from within the Customer Portal section of the site click "Configure" under Portal Home -> Radar in the left hand menu.



Once you have adjusted the settings to your liking, click the "update radar settings" button. Once clicked a notification at the top of the page will say "Radar settings updated successfully" and you will be directed to the "Radar Web Integration" page.

The new settings will now be active with these new edited parameters.

A full description of the 8 parameters is listed in the table below:

Parameter	Description	Default Setting
Timing Options		
Navigation Timing	The delay, in seconds, between the page onLoad event and when Radar records navigation timing.	2 Seconds
Remote Probing	The delay, in seconds, between the page onLoad event and when Radar starts measuring platforms.	2 Seconds
Private Measurements		
Maximum Private Measurements per Page Load	The maximum number of private platforms that Radar will measure per page load.	2
Maximum Private throughput Measurements	The maximum number of throughput measurements of private platforms per page load*	2
Measurement Timeout	The time, in seconds, that Radar waits for a response before recording	2 Seconds

	the test as a failure.	
Public Measurements		
Maximum Community Measurements per Page Load	The maximum number of community platforms that Radar will measure per page load.	4
Maximum Community throughput Measurements	The maximum number of throughput measurements of community platforms per page load.	4

**Note: each user will make 3 HTTP Get Requests per Test cycle (2 x 50Byte, 1x 100Kbyte)*

Note: do not change any values within the JavaScript you have implemented. All control options are managed through the Customer Portal.

A note on Cedexis's data policy

Cedexis's performance measurement and load balancing systems do not collect any PII (Personal Identifiable Information) or user data. They collect anonymous traffic data and/or anonymous location data only. The systems do not set cookies, and do not read cookies.

The Cedexis Website, at http://www.cedexis.com/, and the Cedexis Customer Portal at https://portal.cedexis.com/ have separate data protection statements covering users of the Website and Portal.

Cedexis was recently SaaS Privacy Certified by eTrust as well as granted EU Safe Harbor. If you have any concerns over the privacy of data collected please reach out to your regional team or review http://www.cedexis.com/legal/privacy.html .