Collective2 Data Services API (C2DS)
Revised October 2, 2008

Change log

October 2, 2008

+ Two big additions to the Data Services API: **establishstatus** and **checkin**. These functions allow developers to create applications and services which receive dynamic updates about a C2 users' positions, brokerage fills, subscribed signals, etc.

The paradigm is this. You establish a session, then build a "baseline status" by using the **establishstatus** command for each system you are interested in. Then, periodically, you can use **checkin** to see what, if anything, has changed since your last **checkin**.

Please see **establishstatus** and **checkin** later in this document. (Link)

+ Also: signal nodes now contain reference to the trade IDs (<tradeid>). This was a missing link between **trades** (or what some might call "positions") and **signals**.

In C2's schema, a "trade" consists of a collection of signals for a given instrument. A trade begins (and is assigned a tradeid) when the first opening signal for a given instrument is executed, and the trade continues (with more signals being added to / associated with the trade) until the position finally goes flat. Then, if a new signal subsequently opens a new position, it will belong to a new trade (and have a new tradeid). A signal does not get associated with a trade until it is actually filled by C2.

July 23, 2008

Developers now have access to dynamically resizable system equity charts (p. 22)

&recentclosehours= parameter now available in gettradehistory, to allow you to display only trades completely closed within a certain time frame (p. 7).

Two new commands to allow developers to build "dashboards" that monitor system status more efficiently: **establishstatus** and **checkin** (p. 22)

Delayed quotes (node <quote>) now appear in signal nodes for pending signals, and within open trade nodes. Please note that the P/L calculations for open trades may have been calculated at a different moment in time than the delayed quote. In fact, the delayed quote will always be at least as recent or more recent than the cached P/L calculation for a trade. Therefore, you will probably want to calculate your own P/L calculation using the delayed quote, rather than relying on C2's cached value which is returned in the <result> node.

July 10, 2008

Added XML field <**tradeid**> to signal node in **getsignalhistory** command. This matches the tradeid in the getopenpositions command.

June 26, 2008

getopenpositions (Link)
getpendingsignals (Link)
added optional **&numdays=x** filter to getsignalhistory and gettradehistory

Overview

The goal of the C2 Data Services (C2DS) API is to allow individuals, service providers, independent software vendors (ISVs), and brokers to develop value-added services and functionality on top of the C2 trading system platform. Developers may use C2DS privately, or they may base commercial services upon it.

Ultimately, in later releases, C2DS will allow developers to access all C2 functionality via the API. This includes data retrieval, billing, and subscription management.

Currently – as of the this first, very early, beta release – C2DS allows developers to access almost the entire C2 database via API calls.

What C2DS is… and what it is *not*

C2DS is (currently) a data retrieval service. If you want to write software which analyzes the universe of C2 trading systems, or optimizes customers' portfolios, or calculates statistics about a particular trading system… then you will use C2DS. You will request data which can be processed by your own software, for whatever purposes you can imagine.

> If you want to use automated software to **enter trading signals** into C2, you will not use C2DS. In that case, you use our Signal Entry API. More information about the Signal Entry API can be found here: www.collective2.com/api
>
> If you want to build an AutoTrading client or service – that is, if you want to **pull trade signals out** of C2, in order to pipe them into a brokerage account – then you should use the C2 AutoTrading Interface (C2ATI). To receive C2ATI documentation, please send email to info@collective2.com, along with your C2 user login email, and simply request the C2ATI docs.

Technical Overview

C2DS is simple to use. Your software issues commands via the HTTP protocol's GET command. C2DS returns responses in pseudo-XML. In other words, using the C2DS is exactly like requesting a Web page from a server. That is: if you can write software which requests and parses a Web page, you can use C2DS.

While building and debugging C2DS software, you can use a Web browser. Just enter the commands in the browser's URL, and the C2DS answer will appear on the screen. We recommend using Firefox for this purpose because it displays the returned text properly formatted.

First Concepts: The sessionid

The first step when using C2DS is to request a sessionid. The sessionid is a random string of characters which identifies you and carries within it information about the rights you have to view various data. After you receive your sessionid, you will include it within every request that you make of the C2DS server.

To request a sessionid, you issue a getsessionid command. Here is the content of the GET command. You might want to type this into the browser address line to see what is returned:

```
http://data.collective2.com/cgi-perl/xml.mpl?cmd=getsessionid&userlogin=youremail@email.com&pw=yourC2dataServicesPassword&clientid=softwareName
```

Notice that you need a C2 user account in order to begin a session with the C2DS server. Why is this? It is because C2 needs to know what kind of information we are allowed to show you. For example, if you are subscribed to a trading system, then when you request trade information about a particular system, we will show you open trade information. Non-subscribers will not see this data.

The tying of sessions to particular C2 user identities is a powerful concept. It means that you can create commercial (or non-commercial) service which allows multiple C2 users – not just yourself – to carry out C2 functions. Your software could manage multiple sessions, each with its own sessionid, on behalf of each of your customers.

Notice that, in the example above, we say you need to include your "C2 data Services password" … and not your C2 password. Indeed, if you try to use your general purpose C2 password, you will receive an error. You must use your C2 Data Services password, instead.

What is the C2 Data Services Password? This is a password that each C2 user must set at the same time he gives permission for his account to be used with C2DS. In this way, C2 customers can feel comfortable giving a C2 password out to commercial 3rd-party services and vendors… while knowing that the password he is giving out gives no rights other than data retrieval. So, even if a malicious person discovers your C2DS password, the most he can do is request a list of systems to which you are subscribed. He cannot use the password to log in to C2, to enter trades, or to view personal and credit card information.

(Update July 23, 2008: In order to allow software developers to integrate order-management functionality into their applications, we now give customers the option to let the C2D2 password have trade-entry rights. This means that you can use one password to both monitor positions and enter new trades into systems owned by the customer. Note that the customer must specifically choose to give trade-entry rights to the C2DS password.)

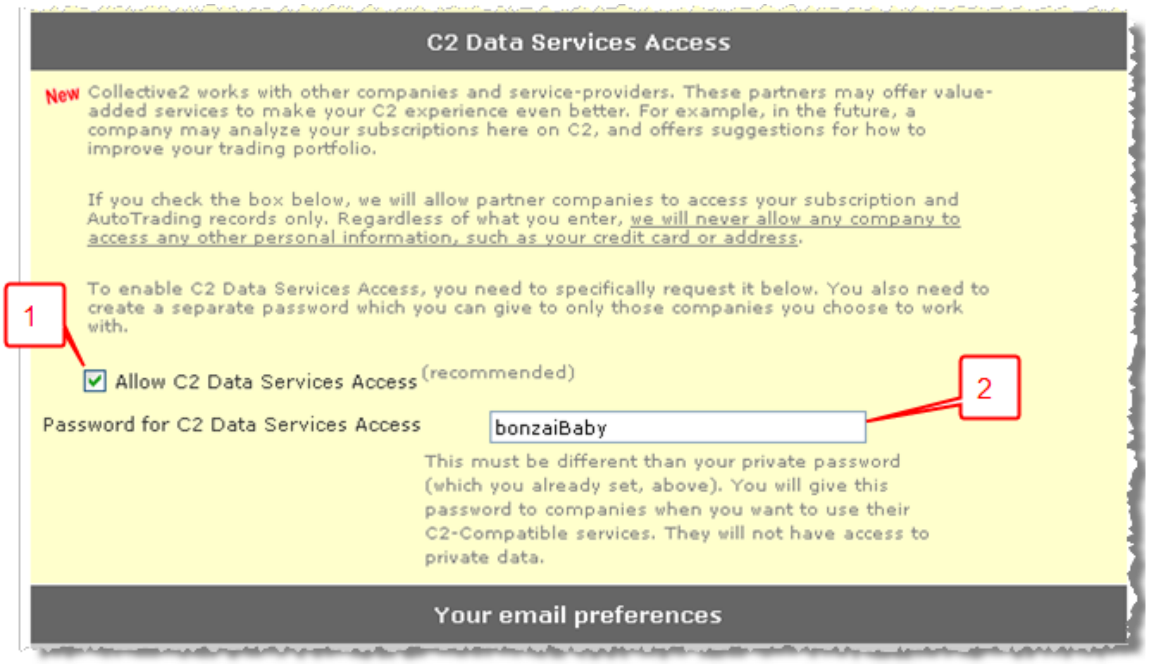
How to set your C2DS password

To set your C2DS password, log in to C2 and click the link on the left-side menu bar that says: "Edit My Account." Alternately, this link will take you there directly:

http://www.collective2.com/cgi-perl/editaccount.mpl

On the screen, you will see an area where you can configure your C2DS password. You need to do two things: (1) explicitly allow C2DS access, and (2) select a password.

Here is a screenshot of the two areas you (or your customers) must fill out in order to use the C2DS:



C2 Data Services Access

New Collective2 works with other companies and service-providers. These partners may offer value-added services to make your C2 experience even better. For example, in the future, a company may analyze your subscriptions here on C2, and offers suggestions for how to improve your trading portfolio.

If you check the box below, we will allow partner companies to access your subscription and AutoTrading records only. Regardless of what you enter, we will never allow any company to access any other personal information, such as your credit card or address.

To enable C2 Data Services Access, you need to specifically request it below. You also need to create a separate password which you can give to only those companies you choose to work with.

☑ Allow C2 Data Services Access (recommended)

Password for C2 Data Services Access: bonzaiBaby

This must be different than your private password (which you already set, above). You will give this password to companies when you want to use their C2-Compatible services. They will not have access to private data.

Your email preferences

Once you allow C2DS access and select a password, you will not need to access this screen again.

Now you can begin to use the C2DS. As we mentioned above, the first step is to request a sessionid. So, using the password selected above, you would issue this command:

http://data.collective2.com/cgi-perl/xml.mpl?cmd=getsessionid&userlogin=youremail@email.com&pw=bonzaiBaby&clientid=algorgasm

> *Note that your software should have a clientid which identifies it to the server. Although you do not need to receive approval from C2 before choosing a clientid or before releasing software, we suggest you choose something that clearly identifies your software. It should be 12 characters or fewer.*

Here is a typical response that you might receive when requesting a sessionid:

```
<c2xml>
  <login>
    <valid>1</valid>
  </login>
  <session>
    <id>uiufUH5NLn2qRVd23CWzHttv9rgaLjp</id>
    <c2userid>50034</c2userid>
    <firstname>Matthew</firstname>
    <lastname>Klein</lastname>
  </session>
  <subsciptions>
    <count>5</count>
    <systemlist>
      <system>
        <systemid>13202557</systemid>
        <systemname>extreme-os</systemname>
      </system>
      <system>
        <systemid>13889808</systemid>
        <systemname>Absolute Returns</systemname>
      </system>
      <system>
        <systemid>23905671</systemid>
        <systemname>VN Forex Club</systemname>
      </system>
      <system>
        <systemid>13202557</systemid>
        <systemname>extreme-os</systemname>
      </system>
    </systemlist>
  <subscriptions>
</c2xml>
```

The response to **getsessionid** contains three main pieces of information. First, it contains your unique sessionid. Note that your sessionid will of course be different than the one above (uiufUH5NLn2qRVHCWMzHttv9rgaLjp). You will need to use this sessionid in all subsequent requests to the C2DS server. Note that sessions id are good for as long as there is activity. However, if too much time passes between requests to the server (currently set at 60 minutes of inactivity) then the session expires and you need to request a new one.

Second, you learn the identity (first name and last name only) of the user whose C2DS password you used.

Finally, the response contains the list of systems to which the user is subscribed.

General XML format

You'll notice that the responses are in well-formed pseudo XML. The main node for every response is called <c2xml>. When naming XML fields (e.g. "systemname", “subscriptions”, etc.) we always chose human sensibility over brevity, even though this costs our server (and yours) bandwidth and disk space. But bandwidth is cheap, and growing more abundant by the day, while your brain is dear, and inexorably shrinking, and so we believe our philosophy of long XML names over short in most cases will be appreciated by programmers.

Summary of All Valid Commands

As of this document draft, the following commands are valid. We assume that anyone who has read this document this far is a programmer by hobby, if not profession, and he will probably want to fire up a browser and simply try the commands to see the response format. For this reason – and also because this is a very early document draft – the documentation that follows is skimpy. We suggest you try a couple examples and figure it out. We’re always available to answer questions, however. We particularly suggest you use the C2 Developer Forum to post your queries, so that other C2 developers can learn from the question and response.

All commands begin with:

http://data.collective2.com/cgi-perl/xml.mpl?

And all commands, except for **getsessionid**, of course, must contain a valid **&session=xxx** parameter, where xxx is the session id that was given to you when you established your session.

The following commands are valid:

http://data.collective2.com/cgi-perl/xml.mpl?**cmd=getsystemroster**&session=xxx

Returns a list of all C2 systems.

---

xml.mpl?**cmd=getsystemoverview&systemid=sss**&session=xxx

Returns summary information about a particular system.

---

xml.mpl?**cmd=getsystemstats&systemid=sss**&session=xxx

Returns a list of all cached statistics and meta-statistics for a particular system.

---

xml.mpl?**cmd=getsubscriptions** &session=xxx

Returns a list of systems to which the user (whose C2DS password is being used) has subscribed.

---

xml.mpl?**cmd=getequityseries&systemid=sss**&session=xxx

Returns a list of system equity (both open and closed positions) over time.

---

xml.mpl?**cmd=getbenchmarkseries&benchmark=SP500&startdate=YYYYMMDD &enddate=YYYYMMDD**&session=xxx

Returns a series of prices of a benchmark. Currently only SP500 index is supported.

---

xml.mpl?**cmd=gettradehistory&systemid=sss**&session=xxx

Returns a list of a system's closed trades. Includes drawdown data. "Trades" in C2 lingo are a unit of measure that may consist of many "signals." A trade starts when a new position is opened, and ends when that position finally becomes flat.

You may add an optional filter **&numdays=DDD** which tells C2 to display only those trades opened within a certain number of days.

To query only recently closed trades, add the following parameter: **&recentclosehours=HHH** where HHH tells C2 to display only those trades completely closed within HHH hours (trades partially closed within that time period will not be displayed.)

---

xml.mpl?**cmd=getsignalhistory&systemid=sss**&session=xxx[&numdays=x]

Returns a list of individual trade signals (already executed). You can specify an optional **&numdays=x** clause, where numdays specifies recency of trades to be displayed (by fill date).

---

xml.mpl?**cmd=getopenpositions&systemid=sss**&session=xxx

Returns a list of a system's currently open trades trades. Includes entry price, entry time, portion of trade that may be closed, and "point value" for futures or forex positions Once a trade is completely closed, it will not appear here.

---

xml.mpl?**cmd=getpendingsignals&systemid=sss**&session=xxx

Returns a list of individual trade signals not yet executed. Includes HTML commentary provided by system vendor, and indicates which trades are conditional upon "parent" signals (a conditional order should not be executed until its parent is executed).

---

xml.mpl?**cmd=logout**&session=xxx

Ends a session, destroys the sessionid, and returns information about the time session was alive and the number of bytes transferred.

---

establishstatus and **checkin**

The goal behind these two features is to allow you to build applications which can receive dynamic updates about all the information a C2 user is interested in – without draining server or client resources, or swamping the network with unimportant data.

The paradigm is this. You use **establishstatus** on a system-by-system basis.

```
http://data.collective2.com/cgi-
perl/xml.mpl?session=xxx&cmd=establishstatus&systemid=29261068
```

For each system user is subscribed to (or offering as a vendor), you'll need to call this once, to build the "baseline" of what's going on in the system. The response will contain virtually everything a user might be interested in regarding the system:

```
<c2xml>
<systemid>29261068</systemid>
<systemname>mvp-3</systemname>
<status>
<timestamp>2008-07-10 16:48:36</timestamp>
  <openPositions>
    <trade>
      <tradeid>33540653</tradeid>
      <symbol>MVV</symbol>
      <instrument>stock</instrument>
      <positionOpened>2008-07-03 12:49:24</positionOpened>
      <quantOpened>580</quantOpened>
      <quantClosed>0</quantClosed>
      <entryPrice>64.18483</entryPrice>
      <pointValue>1</pointValue>
    </trade>
  </openPositions>
  <signalsTradedInOpenPositions>
    <signal>
      <signalid>33521242</signalid>
      <tradeid>33540653</tradeid>
      <filledwhen>2008-07-03 12:49:24</filledwhen>
      <fillprice>64.34</fillprice>
      <action>BTO</action>
      <quant>280</quant>
      <symbol>MVV</symbol>
      <instrument>stock</instrument>
      <marketOrder>1</marketOrder>
      <limitPrice>0</limitPrice>
      <stopPrice>0</stopPrice>
      <TIF>DAY</TIF>
      <postedwhen>2008-07-03 12:50:54</postedwhen>
      <conditionalUpon></conditionalUpon>

<commentary>buy%20MVV%20after%20MZZ%20is%20sold%20and%20stop%20loss%20o
n%20MZZ%20has%20been%20cancelled%3Ccomma%3E</commentary>
    </signal>
    <signal>
      <signalid>33646745</signalid>
      <tradeid>33540653</tradeid>
      <filledwhen>2008-07-10 09:30:41</filledwhen>
      <fillprice>64.04</fillprice>
      <action>BTO</action>
      <quant>300</quant>
      <symbol>MVV</symbol>
      <instrument>stock</instrument>
      <marketOrder>1</marketOrder>
      <limitPrice>0</limitPrice>
      <stopPrice>0</stopPrice>
      <TIF>DAY</TIF>
      <postedwhen>2008-07-09 18:55:55</postedwhen>
      <conditionalUpon></conditionalUpon>

<commentary>PiggyBack%20BUy%20signal%20for%20MVV%0D%0APursuant%20to%20t
```

```
he%20notes%20in%20the%20faq%20file%20I%20have%20scaled%20back%20the%20p
urchase%20so%20the%20program%20is%20using%2090%25%20or%20less%20of%20av
ailable%20margin%2E%0D%0ASee%20notes%20about%201st%20PB%20on%20a%20main
%20buy%0D%0Asee%20more%20detail%20at%20private%20group</commentary>
    </signal>
  </signalsTradedInOpenPositions>
  <pendingSignals>
  </pendingSignals>
  <brokerFillsForOpenPositions>
  </brokerFillsForOpenPositions>
</status>
</c2xml>
```

Now, on a periodic basis (more on this in a moment), you will use **checkin** to see what has changed since your last **checkin** (or, if this is the first **checkin**, then what has changed since your **establishstatus** request):

```
xml.mpl?session=xxx&cmd=checkin&systemid=29261068
```

You need to call this at a reasonable interval for each system. What is a reasonable interval? Well, because this feature is new, and because we want to make sure resources are available for all users, polling should be done no more frequently than once every 7 seconds *for each system in question*. (Later we can make polling more frequent and offer an 'all-in-one' poll for all systems you are interested in.)

The response to **checkin** will look something like this:

```
<c2xml>
<systemid>29261068</systemid>
<systemname>mvp-3</systemname>
<status>
<timestamp>2008-07-10 16:41:50</timestamp>
  <openPositions>
  </openPositions>
  <signalsTradedInOpenPositions>
  </signalsTradedInOpenPositions>
  <pendingSignals>
  </pendingSignals>
  <latestBrokerFills>
  </latestBrokerFills>
  <deltaSinceLast>
    <deltaPosts>
    </deltaPosts>
    <deltaTrades>
    </deltaTrades>
    <deltaKills>
    </deltaKills>
    <deltaExpires>
    </deltaExpires>
  </deltaSinceLast>
```

```
</status>
</c2xml>
```

Notice how the XML above shows only the <u>deltas</u> since the last **checkin**. The example above is pretty meaningless: it means nothing at all has changed since you last checked in.

Note, too, that real live AutoTrade broker fill data is included in here. There's some limitations for now... data for Gen3 brokers can be up to 15 minutes delayed. (Gen1 and 2 is nearly instantaneous.)

Important: If you don't establishstatus **in a given xml session, then** checkin **won't work.**

Also, please be kind: While you can re-build your baseline data by polling for establishstatus **more than once, but it's an** *exceedingly* **expensive operation in terms of server usage, so please only use infrequently.**

Detailed Response Examples

We recommend you try for yourself issuing the commands listed above, and exploring the responses. Because the data that is returned can sometimes be large, we necessarily need to excerpt it in the examples below.

cmd=getsystemroster

```
c2xml>
  <systemroster>
    <system>
      <id>555</id>
      <name>Test System</name>
      <agedays>701.39</agedays>
      <numsigs>1</numsigs>
      <startcash>100000</startcash>
      <win>2770</win>
      <loss>0</loss>
      <c2userid>250</c2userid>
    </system>
    <system>
      <id>666</id>
      <name>AT Research Test</name>
      <agedays>719.44</agedays>
      <numsigs>5</numsigs>
      <startcash>100000</startcash>
      <win>18586.5</win>
    <loss>0</loss>
    <c2userid>250</c2userid>
  </system>
  <system>
    <id>97600</id>
    <name>The Cadbury System</name>
    <agedays>1401.56</agedays>
    <numsigs>25</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>97571</c2userid>
  </system>
  <system>
    <id>147249</id>
    <name>Fundamental Bond Futures</name>
    <agedays>1391.44</agedays>
    <numsigs>0</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>147154</c2userid>
  </system>
  <system>
    <id>167688</id>
    <name>The Four O'Clock Trade</name>
    <agedays>1388.56</agedays>
    <numsigs>2</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>167615</c2userid>
  </system>
  <system>
    <id>179985</id>
    <name>Intelligent Market Timing</name>
    <agedays>1387.15</agedays>
    <numsigs>8</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>179883</c2userid>
  </system>
  <system>
    <id>182188</id>
    <name>Leader in an accelerating group</name>
    <agedays>1387.00</agedays>
    <numsigs>3</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>182059</c2userid>
  </system>
  <system>
    <id>188582</id>
    <name>Random Zen Master</name>
    <agedays>1386.35</agedays>
    <numsigs>2</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>97571</c2userid>
  </system>
  <system>
    <id>190324</id>
    <name>OptionSniper</name>
    <agedays>1386.17</agedays>
    <numsigs>0</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>190277</c2userid>
  </system>
  <system>
    <id>192185</id>
    <name>The Option Game</name>
    <agedays>1386.07</agedays>
    <numsigs>0</numsigs>
    <startcash>100000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>192095</c2userid>
  </system>
  <system>
    <id>204639</id>
    <name>The Taleb Factor</name>
    <agedays>1384.53</agedays>
    <numsigs>36</numsigs>
    <startcash>100000</startcash>
    <win>2367.36</win>
    <loss>47050</loss>
    <c2userid>204606</c2userid>
  </system>
```

………excerpted………

```
   <system>
    <id>26915699</id>
    <name>qntest</name>
    <agedays>0.20</agedays>
    <numsigs>0</numsigs>
    <startcash>10000</startcash>
    <win>0</win>
    <loss>0</loss>
    <c2userid>26915670</c2userid>
  </system>
  <system>
    <id>26916009</id>
    <name>#1 Forex</name>
    <agedays>0.16</agedays>
    <numsigs>1</numsigs>
    <startcash>25000</startcash>
    <win>0</win>
    <loss>175.01</loss>
```

```
      <c2userid>26882267</c2userid>
    </system>
    <system>
      <id>26916492</id>
      <name>YM Mini-Dow</name>
      <agedays>0.10</agedays>
      <numsigs>0</numsigs>
      <startcash>100000</startcash>
      <win>0</win>
      <loss>0</loss>
      <c2userid>26895464</c2userid>
    </system>
    <system>
      <id>26917301</id>
      <name>The Prophet Forcaster</name>
      <agedays>0.04</agedays>
      <numsigs>0</numsigs>
      <startcash>100000</startcash>
      <win>0</win>
      <loss>0</loss>
      <c2userid>26729849</c2userid>
    </system>
  </systemroster>
  </c2xml>
```

cmd=getsystemoverview&systemid=sss

```
<system>
  <systemid>13889808</systemid>
  <systemname>Absolute Returns</systemname>
  <startedwhen>2007-06-13 18:56:49</startedwhen>
  <creator>
    <c2userid>50034</c2userid>
    <name>Matthew Klein</name>
    <org>Matthew Klein LLC, Inc.</org>
  </creator>
  <businessmodel>
    <modelnum>1</modelnum>
    <modelname>periodic</modelname>
    <periodname>quarter<periodName>
    <freetrial>0</freetrial>
    <charge>100</charge>
  </businessmodel>
  <tradestyle>
    <stockpcnt>0.008</stockpcnt>
    <optionpcnt>0.008</optionpcnt>
    <futurepcnt>0.934</futurepcnt>
    <forexpcnt>0.049</forexpcnt>
    <avgTradeMinute>70</avgTradeMinute>
  </tradestyle>
  <summarystats>
    <startcapital>75000</startcapital>
    <openpl>2090</openpl>
    <closedpl>278.25</closedpl>
    <marginused>29135</marginused>
    <equity>32890</equity>
    <cash>62834.9</cash>
    <trades>61</trades>
    <numloss>39</numloss>
    <dollarwin>4207.9</dollarwin>
    <dollarloss>1513</dollarloss>
  </summarystats>
  <charts>

<small>http://www.collective2.com/images/charts/tiny/c13889808.png</sma
ll>

<big>http://www.collective2.com/images/charts/equitychart13889808.png</
big>
  </charts>
  <descrip>
    <short></short>
    <long></long>
  </descrip>
</system>
```

---

cmd=getsystemstats&systemid=sss

```
<c2xml>
<systemid>13889808</systemid>
<systemname>Absolute Returns</systemname>
<stats>
  <stat>
    <name>anyshorts</name>
    <val>1</val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
  <stat>
    <name>avgadversedd</name>
    <val>-0.13</val>
    <calced>2007-06-26 20:30:10</calced>
  </stat>
  <stat>
    <name>avgpips</name>
    <val></val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
  <stat>
    <name>avgptsfutures</name>
    <val>0.108</val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
  <stat>
    <name>avgptsstocks</name>
    <val>2.39</val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
  <stat>
  <name>avgpts_@ER2</name>
  <val></val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>avgpts_@ES</name>
  <val>0.108</val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>avgpts_@SP</name>
  <val></val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>avgpts_@US</name>
  <val></val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>avgpts_@YM</name>
  <val></val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>avgtradepl</name>
  <val>5.39</val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>forexpercent</name>
  <val>0.049</val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>futurespercent</name>
  <val>0.934</val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>holdandhope</name>
  <val>0.437</val>
  <calced>2007-06-26 19:52:41</calced>
</stat>
<stat>
  <name>j1YrRisk95</name>
  <val></val>
  <calced>2007-06-26 08:50:18</calced>
</stat>
<stat>
  <name>j1YrRiskExpShort</name>
  <val></val>
  <calced>2007-06-26 08:50:18</calced>
</stat>
<stat>
  <name>jAnnReturn</name>
  <val>0.178</val>
```

```
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jCalmar</name>
    <val>317.852</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jDDQuartMean4</name>
    <val>0.001</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jRetnQuartMean1</name>
    <val>1</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jSharpe</name>
    <val>6.254</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jSharpeConfH</name>
    <val>17.439</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jSharpeConfL</name>
    <val>-4.271</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jSortino</name>
    <val>28.231</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>jVAR95</name>
    <val></val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>latesttradedaysago</name>
    <val>0.1</val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
  <stat>
    <name>lowEquityAmt</name>
    <val>74837</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>lowEquityDate</name>
    <val>20070614</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>maxadversedd</name>
    <val>-0.23</val>
    <calced>2007-06-26 20:30:10</calced>
  </stat>
  <stat>
    <name>maxdrawdown</name>
    <val>0</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>maxdrawdownend</name>
    <val>20070614</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>maxdrawdownPcnt</name>
    <val>0.22</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>maxdrawdownstart</name>
    <val>20070613</val>
    <calced>2007-06-26 08:50:18</calced>
  </stat>
  <stat>
    <name>optionpercent</name>
    <val>0.008</val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
  <stat>
    <name>stockpercent</name>
    <val>0.008</val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
  <stat>
    <name>trades</name>
    <val>Mostly Futures; (some stocks, forex, options)</val>
    <calced>2007-06-26 19:52:41</calced>
  </stat>
</stats>
</c2xml>
```

---

cmd=getequityseries&systemid=sss

Here we do abbreviate XML nodes in order to save some bandwidth. The inscrutable abbreviations below mean:

E=equity point

T=timestamp that equity snapshot was taken (these are irregularly spaced throughout the day.

O=open position equity

C=Closed positions / cash

```
<c2xml>
<systemid>13889808</systemid>
<equityseries>
  <e>
    <t>2007-06-13 18:56:49</t>
    <o>0</o>
    <c>75000</c>
  </e>
  <e>
    <t>2007-06-14 01:08:57</t>
    <o>-12</o>
    <c>75000</c>
  </e>
  <e>
    <t>2007-06-14 09:22:47</t>
    <o>0</o>
    <c>74838</c>
  </e>
  <e>
    <t>2007-06-14 17:41:04</t>
    <o>0</o>
    <c>75237</c>
  </e>
  <e>
    <t>2007-06-14 22:53:51</t>
    <o>0</o>
    <c>75237</c>
  </e>
  <e>
    <t>2007-06-15 07:04:03</t>
    <o>0</o>
    <c>75237</c>
  </e>
  <e>
    <t>2007-06-15 17:10:20</t>
    <o>0</o>
    <c>75112</c>
  </e>
  <e>
    <t>2007-06-15 21:31:25</t>
    <o>0</o>
    <c>75112</c>
  </e>
  <e>
    <t>2007-06-16 04:38:29</t>
    <o>0</o>
    <c>75112</c>
  </e>
  <e>
    <t>2007-06-16 13:25:29</t>
    <o>0</o>
    <c>75112</c>
  </e>
                ………excerpted…………
</equityseries>
</c2xml>
```

---

cmd=getbenchmarkseries&benchmark=SP500&startdate=20070501&enddate=20070507

B = benchmark data point

D = date

P = price

```
<c2xml>
<benchmark>SP500</benchmark>
<benchmarkseries>
  <b>
    <d>20070501</d>
    <p>1486.3</p>
  </b>
  <b>
    <d>20070502</d>
    <p>1495.92</p>
  </b>
  <b>
    <d>20070503</d>
    <p>1502.39</p>
  </b>
  <b>
    <d>20070504</d>
    <p>1505.62</p>
  </b>
  <b>
    <d>20070505</d>
    <p>1505.62</p>
  </b>
  <b>
    <d>20070506</d>
    <p>1505.62</p>
  </b>
  <b>
    <d>20070507</d>
    <p>1509.48</p>
  </b>
</benchmarkseries>
</c2xml>
```

---

cmd=gettradehistory&systemid=sss

```
<c2xml>
<systemid>13889808</systemid>
<systemname>Absolute Returns</systemname>
<trades>
  <trade>
```

```
    <tradeid>26752257</tradeid>
    <type>future</type>
    <earliestopen>2007-06-13 19:58:17</earliestopen>
    <openaction>STO</openaction>
    <openquant>1</openquant>
    <opensymbol>@ESU7</opensymbol>
    <openprice>1530.0000</openprice>
    <lastclose>2007-06-13 20:03:12</lastclose>
    <closeaction>BTC</closeaction>
    <closequant>1</closequant>
    <closeprice>1529.7500</closeprice>
    <maxdrawdown>
    </maxdrawdown>
  </trade>
  <trade>
    <tradeid>26752302</tradeid>
    <type>future</type>
    <earliestopen>2007-06-13 20:03:23</earliestopen>
    <openaction>BTO</openaction>
    <openquant>1</openquant>
    <opensymbol>@ESU7</opensymbol>
    <openprice>1529.7500</openprice>
    <lastclose>2007-06-13 20:28:33</lastclose>
    <closeaction>STC</closeaction>
    <closequant>1</closequant>
    <closeprice>1529.5000</closeprice>
    <maxdrawdown>
      <amt>-37.5</amt>
      <openquant>1</openquant>
      <entryprice>1529.75</entryprice>
      <worstprice>1529</worstprice>
      <maxddwhen>2007-06-13 20:21:00</maxddwhen>
      <calcedwhen>2007-06-20 18:22:21</calcedwhen>
    </maxdrawdown>
  </trade>
  <trade>
    <tradeid>26752609</tradeid>
    <type>future</type>
    <earliestopen>2007-06-13 20:28:43</earliestopen>
    <openaction>STO</openaction>
    <openquant>1</openquant>
    <opensymbol>@ESU7</opensymbol>
    <openprice>1529.5000</openprice>
    <lastclose>2007-06-13 20:41:01</lastclose>
    <closeaction>BTC</closeaction>
    <closequant>1</closequant>
    <closeprice>1529.5000</closeprice>
    <maxdrawdown>
    </maxdrawdown>
  </trade>
  <trade>
    <tradeid>26752751</tradeid>
    <type>future</type>
    <earliestopen>2007-06-13 20:41:11</earliestopen>
    <openaction>BTO</openaction>
    <openquant>1</openquant>
    <opensymbol>@ESU7</opensymbol>
    <openprice>1529.5000</openprice>
    <lastclose>2007-06-13 22:34:29</lastclose>
    <closeaction>STC</closeaction>
    <closequant>1</closequant>
    <closeprice>1529.2500</closeprice>
    <maxdrawdown>
      <amt>-12.5</amt>
      <openquant>1</openquant>
      <entryprice>1529.5</entryprice>
      <worstprice>1529.25</worstprice>
      <maxddwhen>2007-06-13 22:01:00</maxddwhen>
      <calcedwhen>2007-06-20 18:22:18</calcedwhen>
    </maxdrawdown>
  </trade>

          ………excerpted………

  <trade>
    <tradeid>26913791</tradeid>
    <type>option</type>
    <earliestopen>2007-06-26 15:52:30</earliestopen>
    <openaction>STO</openaction>
    <openquant>10</openquant>
    <opensymbol>IOCGH</opensymbol>
    <openprice>1.2000</openprice>
    <lastclose></lastclose>
    <closeaction>BTC</closeaction>
    <closequant></closequant>
    <closeprice></closeprice>
    <maxdrawdown>
    </maxdrawdown>
  </trade>
  <trade>
    <tradeid>26913804</tradeid>
    <type>stock</type>
    <earliestopen>2007-06-26 15:53:17</earliestopen>
    <openaction>BTO</openaction>
    <openquant>1000</openquant>
    <opensymbol>IOC</opensymbol>
    <openprice>27.9400</openprice>
    <lastclose></lastclose>
    <closeaction>STC</closeaction>
    <closequant></closequant>
    <closeprice></closeprice>
    <maxdrawdown>
    </maxdrawdown>
  </trade>
</trades>
</c2xml>
```

---

xml.mpl?**cmd=gesignalhistory&systemid=sss**&session=xxx

```
<c2xml>
```

```
<systemid>13889808</systemid>
<systemname>Absolute Returns</systemname>
<signals>
  <signal>
    <sigid>26752247</sigid>
    <posted>2007-06-13 19:57:06</posted>
    <traded>2007-06-13 19:58:17</traded>
    <action>STO</action>
    <quant>1</quant>
    <symbol>@ESU7</symbol>
    <type>future</type>
    <market>0</market>
    <limit>0</limit>
    <stop>1529</stop>
    <tif>DAY</tif>
    <tradeprice>1530</tradeprice>
    <comment></comment>
  </signal>
  <signal>
    <sigid>26752274</sigid>
    <posted>2007-06-13 20:00:09</posted>
    <traded>2007-06-13 20:03:12</traded>
    <action>BTC</action>
    <quant>1</quant>
    <symbol>@ESU7</symbol>
    <type>future</type>
    <market>0</market>
    <limit>0</limit>
    <stop>1529.75</stop>
    <tif>DAY</tif>
    <tradeprice>1529.75</tradeprice>
    <comment></comment>
  </signal>
  <signal>
    <sigid>26752269</sigid>
    <posted>2007-06-13 20:00:09</posted>
    <traded>2007-06-13 20:03:23</traded>
    <action>BTO</action>
    <quant>1</quant>
    <symbol>@ESU7</symbol>
    <type>future</type>
    <market>0</market>
    <limit>0</limit>
    <stop>1529.75</stop>
    <tif>DAY</tif>
    <tradeprice>1529.75</tradeprice>
    <comment></comment>
  </signal>
  <signal>
    <sigid>26752593</sigid>
    <posted>2007-06-13 20:28:05</posted>
    <traded>2007-06-13 20:28:33</traded>
    <action>STC</action>
    <quant>1</quant>
    <symbol>@ESU7</symbol>
    <type>future</type>
    <market>0</market>
    <limit>0</limit>
    <stop>1529.5</stop>
    <tif>DAY</tif>
    <tradeprice>1529.5</tradeprice>
    <comment></comment>
  </signal>
  <signal>
    <sigid>26752590</sigid>
    <posted>2007-06-13 20:28:05</posted>
    <traded>2007-06-13 20:28:43</traded>
    <action>STO</action>
    <quant>1</quant>
    <symbol>@ESU7</symbol>
    <type>future</type>
    <market>0</market>
    <limit>0</limit>
    <stop>1529.5</stop>
    <tif>DAY</tif>
    <tradeprice>1529.5</tradeprice>
    <comment></comment>
  </signal>

        ………excerpted………

  <signal>
    <sigid>26914865</sigid>
    <posted>2007-06-26 17:36:05</posted>
    <traded>2007-06-26 17:36:05</traded>
    <action>STC</action>
    <quant>3</quant>
    <symbol>EURUSD</symbol>
    <type>forex</type>
    <market>1</market>
    <limit>0</limit>
    <stop>0</stop>
    <tif>DAY</tif>
    <tradeprice>1.3455</tradeprice>
    <comment></comment>
  </signal>
</signals>
</c2xml>
```

---

xml.mpl?**cmd=logout**&session=xxx

```
<c2xml>
  <logout>
    <minutes>82.8</minutes>
    <bytes>1145766</bytes>
  </logout>
</c2xml>
```

Error examples:

Unhandled command:

```
<c2xml>
  <error>1</error>
  <message>Unhandled cmd (getoogaboogadata)</message>
</c2xml>
```

Unknown system:

```
<c2xml>
  <error>1</error>
  <message>Unknown systemid (2221111)</message>
</c2xml>
```

Charting

Application developers have asked for access to C2's dynamic charting features. You can display a system equity chart as follows:

```
http://www.collective2.com/cgi-perl/ycharts.mpl?sid=xxx
```

Where sid= is the system id number. Typing the URL above in your browser's address area will send a .png image to the browser. Thus, to contain the image within an HTML page, your HTML would contain:

```
<img src="http://www.collective2.com/cgi-perl/ycharts.mpl?sid=xxx">
```

The chart defaults to C2's standard 375 wide by 250 tall, and includes the system's name rasterized within the image. You can change with and height by adding **&width=W** and **&height=H** parameters to the URL. Note that the image format has been optimized for 375 X 250, and – while it will display well and look right through a range of dimensions, as you move towards the smaller limits, the layout, text, and scale of the axes will begin to look funky.

Retrieving new information quickly: commands *establishstatus* and *checkin*

Some developers have expressed an interest in building all-encompassing applications that act as a kind of live "dashboard" for Collective2 users – displaying positions, trades, ever-changing P/L's – even a user's autotrade fills! – etc.

To aid the development of these type of applications, C2 had created two new very-efficient commands: **establishstatus** and **checkin**.

The idea behind these commands is that C2 wants to simplify the retrieval, within one polling cycle, of all of the information a particular C2 customer has access to and has interest in.

The methodology of using these commands is as follows. You use **establishstatus** to retrieve baseline information for a particular system. Then, at a reasonable frequency (more on this in a moment), you use the **checkin** command to find out what has changed since the last poll. (The first checkin will reveal what, if anything, has changed since the establishstatus command, and all subsequent checkins will reveal what has changed since the prior checkin.)

These commands must be called on a system by system basis. Example:

```
http://data.collective2.com/cgi-
perl/xml.mpl?session=xxx&cmd=establishstatus&systemid=sss
```

or

```
http://data.collective2.com/cgi-
perl/xml.mpl?session=xxx&cmd=checkin&systemid=sss
```

Earlier, I alluded to thefact that you should use these commands at a reasonable frequency. While they have been designed to be sprightly and not overburden C2 infrastructure, there is some cost for each polling cycle. For this reason, we request that you never poll more frequently than once every 10 seconds for any particular system. (This means you may poll for system1 and then system2 without any delay, but that at least 10 seconds must elapse between polls for system1.) C2 will enforce this polling delay by caching results for at least 10 seconds. Therefore, there will be no point in polling more frequently than this.

Conclusion

We hope that software developers will be pleased with this first iteration of the C2DS. Shortly we will be rolling out additional functionality, including retrieval of autotrade and slippage data, subscription (and subscription resale) billing, and subscriber management.

Please share your feedback with us, and feel free to ask questions. We want to help you build great C2-Compatible software! Although we prefer public posts in the Collective2 C2 Software Development forum, we are also available privately via email at info@collective2.com.