



# Building a Mobile 2.0 World: Information and Process Sharing within Wireless Ad-Hoc Communities

By Bob DeAnna, Chief Technology Officer

*Gartner predicts a future of mobile 'ambient business', where consumers explore their environment to find relevant value propositions...Data will be selectively pushed to the user based on context, matching the users' needs, interests, mood, location and even recent behavior...Location, combined with personal presence, creates transient communities, which offers an opportunity for better use of [wireless] time or marketing.*

*-- Nick Jones, Gartner Says Corporate Mobility Becomes Mainstream and Outlines the Shape of the Future with Mobile Business 2.0, April 3, 2006*

## Executive Summary

IT organizations must prepare for the challenges and opportunities presented by the explosion of wireless and embedded devices within the workforce and on the “playing field”. Mobile 2.0 solutions are often hyped, but are they truly different from just mobilizing the web? Does a platform exist that is powerful enough to bring the next generation of intelligent, distributed, transactional, interoperable applications to the edge?

Read this whitepaper to learn how to identify and utilize a platform that allows the sharing of information and processes across multiple devices including Smartphones, ultra mobile PC's, sensors, RFID readers, and servers. Dynamic gaming, RFID, and multi-party workflow scenarios are covered. Also, learn how the Voyager Edge™ Intelligent Distributed Computing Platform can be leveraged in leading edge software today, as well as scenarios and code samples that will stir the imagination towards building the applications of the future.

## Contents

Envisioning a Mobile 2.0 World.....	2
Scenarios of the Mobile 2.0 World .....	2
Mobile 2.0 Platform Requirements: A Top 10 List .....	5
The Mobile 2.0 Super Highway.....	7
Sample Application .....	8
Creating a Mobile 2.0 World Today .....	10

---

### Envisioning a Mobile 2.0 World

Much ado has been made about Mobile 2.0 applications of the future, however finding apps that truly revolutionize the way we live can be difficult to identify. True Mobile 2.0 applications will be pervasive and impact every aspect of our lives...how we work, learn, communicate, receive medical care, shop, play, protect ourselves and defend our nation. These applications will open up a new world of business opportunities for the entrepreneur and established corporations alike.

To make this vision a reality, Mobile 2.0 applications must be powered by a next generation platform that will be able to operate on many wireless/mobile networks, devices, and embedded systems.

Applications must seamlessly interoperate with platforms, be it JEE, .NET or a legacy system, that provide both high speed and SOAP-based service oriented architectures when developing/ planning solutions for today and tomorrow. This platform will turn the devices of the mobile professional or consumer into wireless learning machines that know the user's behaviors and habits. Based on the user's state (location, work or play, time/day, etc) these devices will be able to deduce the user's needs and proactively deliver information without requiring the user to 'Google' for, or even think about.

The next evolution of the Mobile World that we are about to enter will allow the "edge" devices, be they business or consumer, human or machine operated, to participate in coordinated knowledge sharing, problem

solving, and transactions that span dynamically assembled communities of devices, and multiple companies/ organizations and/or consumers. It will enable a whole new world of micro commerce through location-based, extreme personalization, intelligence, and cognitive analysis.

Those familiar with mainstream platforms may be surprised that a platform to build such applications **exists today** and has been available for more than 10 years. Over the years, Recursion Software's Voyager Edge™ Intelligent Distributed Computing Platform has offered technology ahead of its time– yet now perfectly suited for a Mobile 2.0 world.

Lets take a closer look at several Mobile 2.0 sample scenarios and discuss the role that agent-based applications will play in revolutionizing communications as well as the requirements of the software platform needed.

### Scenarios of the Mobile 2.0 World

A true Mobile 2.0 world involves the interaction between people and dispersed, intelligent wireless and embedded edge devices that are at any given moment in proximity to each other and/or are participants in a shared process. Each person or device will benefit from the sharing of knowledge and interpretation of the events going on around them and form communities to disseminate information.

The following is a brief outline some Mobile 2.0 scenarios that are within our reach

today, beginning with fun, every day scenarios, and progressing to scenarios that will dictate our health and safety in the post 9/11 world. Examples range from a home/office, a shopping mall, and an amusement park, to an accident scene, a secure boarder, and a battlefield.

[Read about additional scenarios and third-party agent applications by visiting Recursion's white paper library at [recursionsw.com/About\\_Us/wp\\_mobileagents.html](http://recursionsw.com/About_Us/wp_mobileagents.html).]

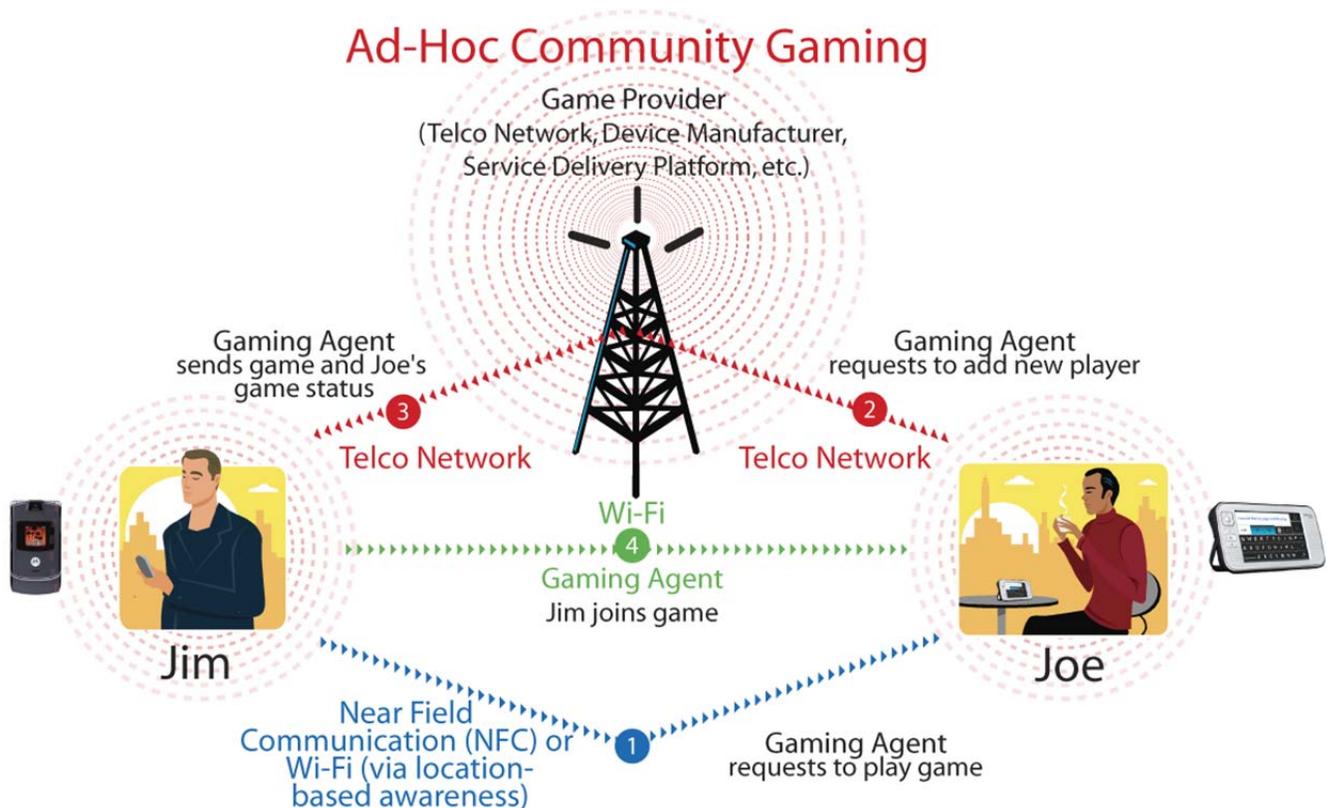
### Home Entertainment Agent

Scaled-down systems exist today, however they do not offer the full capabilities that agent-based applications bring to the table. Imagine the convenience of having a Smartphone that can remotely control and seamlessly interact with the devices in a user's home or office. Suddenly, forgetting to record "the big game" is no longer an issue, but becomes a matter of setting

controls via a Smartphone. What more, a user could set music, lighting and climate controls to their preference on before arrival.

### Interactive Gaming 2.0

Gaming on a Smartphone no longer needs to be a solitary activity, but could be shared within a community of other players, whether on the subway, in a café, or anywhere imaginable. Using a *Gaming Agent Manager*, a user can send a game request from one device to another, or proactively have the device search for an active game to join. The request is then forwarded wirelessly to the commercial entity providing the game, and in turn, the game's status and software is forwarded to the requestor. The new gamer is now able to play the same game against other enthusiasts within the game's dynamically formed community.



### **A Virtual Office at Your Fingertips**

Unlike simply voice and email notifications a *Desktop Computer Software Agent* can send a notification if the user has received an urgent fax or email and can dial out on the phone to retrieve the fax/email to the Smartphone. If the user is driving, he/she can request to convert the data to an audio stream that can be played on the car's sound system (conversion software permitting). If the fax/ email is deemed important, the user can forward it to a colleague and request a hard copy be forwarded to his/her hotel (which can also be booked through the agent), so that it is waiting when the user arrives.

### **Business Intelligence**

In a Mobile 2.0 world, files, processes and information can be shared from any device with anyone at anytime. It is not hard to imagine applications containing *Information Workflow Agents* that can coordinate information sharing between multiple parties and their wireless devices, regardless of what devices they use (Smartphones, PDA's, desktops, laptops, etc.). Think of possible scenarios, particularly in the world of insurance claims, or patient billing, or even organ donation. When information sharing (claims/bill adjustments/complaints or organ availability) breaks down, processing is delayed, and valuable, sometimes life threatening time is lost. If current status and other relevant information can be transferred to all the pertinent parties and to all of their devices, the time to resolution can be reduced dramatically.

### **Healthcare & Emergency Response**

A *Personal Healthcare Agent* application running on a smart-phone could carry an encrypted version of a Personal Health Record. The PHR contains a list of allergies, pre-conditions, current medications, recent medical history, etc. If a user are in an accident or need emergency care, the first responder can wirelessly access the user's health information. This information could then be passed on wirelessly to the hospital

emergency room where the patient is to be admitted.

### **Security**

These same capabilities will also help people in times of crises and stress, save lives at home, protect our borders, and provide a quantum step forward in the effectiveness of soldiers on the battlefield. For securing our homes, offices and national borders, *Security Software Agents* of various types will be running on stationary or mobile cameras, video recorders, and sensors, the PDAs and Smartphones of our Security Border Patrollers, and centralized monitoring locations. Additionally, cameras, video records, and sensors may be located on unmanned autonomous vehicles (UAVs) and robots. These intelligent devices can filter and analyze information as data is gathered and can communicate to owner's/patroller's handheld devices, monitoring center(s) for archival, review, (facial) recognition processing, etc. Multiple micro-events (such as a door opening without key, a person entering a building, a person not identified can be analyzed by collaborating software agents on other nearby devices to determine a complex event (intruder has entered) and communicate such information to appropriate personnel regardless of the device at their disposal. Additionally, other devices can receive information from owner/monitor center(s) to manage home/office/border functions while border police or patrollers are dispatched.

A final extremely compelling scenario is using this next-generation technology to create an intelligent first-responder/military squadron with minimal need for human combatants in the world's most dangerous and hazardous areas. The combination of *Situational Awareness Agents* managing the operations of individual UAV's and robots, as well as groups of them, will be the first responders, terrorist combatants and freedom fighters of tomorrow. These software agents, processing information

from various co-located sensors in real time, will have cognitive capabilities that allow them to process varied micro-events/data and determine the complex events/conditions occurring around them. These events/conditions can be military, biochemical, or geophysical in nature. Agents will collaborate over varied wireless networks including Ultra Wide Band. This will occur not only across dynamically formed groups, but also with human managers/commanders and other field personal to determine a situational awareness that is extremely difficult to do for humans to do when placed in extremely dangerous and chaotic conditions.

Imagine a multi-scenario that combines Healthcare, Emergency Response, and Security and the efficiencies that can be achieved within a persistent, reliable community of devices. In a post-911 world, efficiencies can equal lives saved.

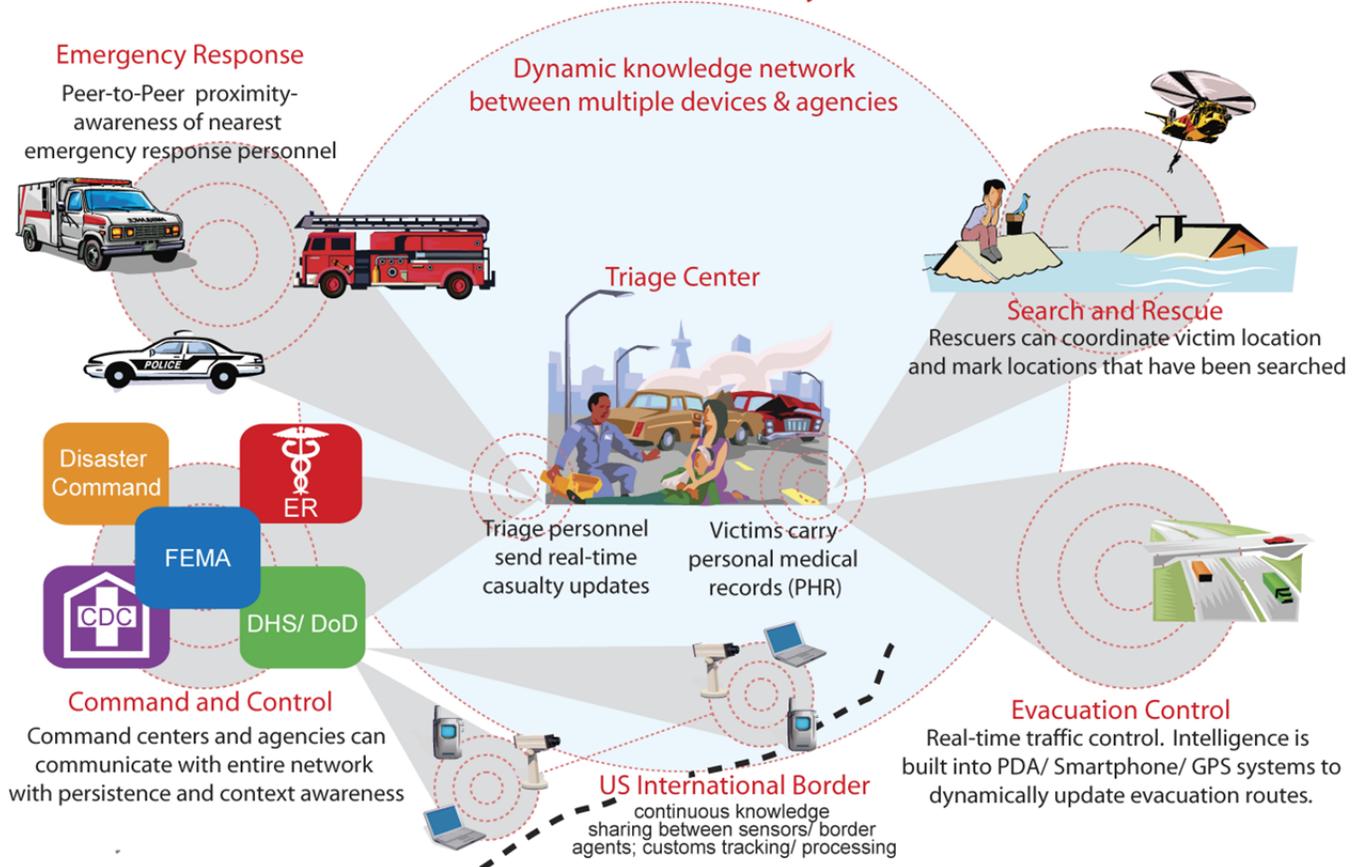
## Mobile 2.0 Platform Requirements: A Top 10 List

After reviewing the previous scenarios, a Top 10 list of features necessary in a software platform that can achieve the above applications warrants consideration.

### 1. Wireless/Embedded Device Support

A true Mobile 2.0 application must be able to run, in some form, on all devices. To do so, the next-generation platform must be pervasive and supported on these same devices. Software agents comprising these applications, also need to be able to run in the popular embedded software stacks such as Java's Micro Edition, Microsoft's Compact Framework, and OSGI Containers on a wide range of embedded operating systems such as Windows Mobile, Symbian, and Embedded Linux and Sun's JavaFX to name just a few.

## Multi-Device Community Scenario



## **2. Decentralized and Centralized Messaging w/Ad-Hoc Communities**

A true Mobile 2.0 application must be able to communicate to groups of devices/systems without the need of a centralized messaging server. It must support the creation of ad-hoc communities of devices or nodes, as well as the ability to support filtering of messages across these communities.

To effectively accomplish this, support must exist for devices joining and leaving the network, which will result in changing "internet" addresses. Integration with a SIP Server is a must, not only provide dynamic IP support, but voice, data, and chat support over IP. The platform must also support passing messages over standard centralized messaging servers for integration with enterprise and legacy systems. More specifically, there needs to be seamless integration with Microsoft's Message Queue (MSMQ), Java's Message Server (JMS) and Object Management Group (OMG)'s Data Description Service (DDS), etc.

## **3. Mobile SOA Architecture**

Next generation applications must provide a Service-Oriented Architecture (SOA). These services need to be accessible via centralized Web Service Container such as Microsoft's Internet Information Services (IIS) and those provided in the Java world, but also must be accessible in a decentralized fashion directly to agents, exposed as web services, that are running on edge and wireless devices. The location of intelligent mobile agents and the Mobile Web Services they expose must be irrelevant to the web service client. Finally, all agents need to be accessible by a Service Description in a Yellow-pages directory, ideally one that is Universal Description, Discovery, and Integration (UDDI) compliant.

## **4. Increase Network Survivability and Mobility**

Mobile 2.0 applications must be able to process data at the source to minimize network traffic, handle unreliable and/or limited network connections, and adjust to hardware failures or CPU load. Therefore, these devices must be able to persist data via a micro database. Additionally, the software components or agents running on edge devices need to support multiple wireless protocols (GSM, CDMA, Wi-Fi, UWB, Bluetooth, NFC, RFID etc) and associated networks (Telco, Wide Area, Local, Personal etc). Ideally, they will dynamically reconfigure themselves to use a communication protocol that best matches the capabilities of their current network connection and the current node(s) they are in communication with.

## **5. Security**

In the Mobile 2.0 world, it will often be necessary to maintain data on edge devices. The security concerns facing enterprises today will need to incorporate solutions that extend into the Mobile 2.0 environment. Applications must provide an extremely high level of security to ensure privacy and protection from rogue/viral clients and software agents. This will involve security agents and agent managers that provide capabilities above and beyond the current encryption, authentication, and authorization that are currently employed in today's centralized client server applications.

## **6. Provide a Single Unified Platform for .NET, Java and C++**

These Mobile 2.0 applications cannot be limited to a single development environment and programming language. Furthermore, the exact same API should be provided to .NET, Java and C++ developers. This would greatly increase programmer productivity and allow developers from all "camps" to easily work together and share software.

## 7. Seamless .NET, Java and Legacy Interoperability

Next generation applications must seamlessly integrate with .NET, JEE and legacy (MVS, CORBA, etc.) enterprise systems and services, and any combination thereof, in either a traditional Web Services architecture, or in a high performance manner. These Mobile 2.0 applications may often need to communicate with more than one enterprise or organization.

## 8. Transactional at the Edge

Next generation applications must extend transactions from the enterprise to include edge devices to allow for distributed, but coordinated tasks among peers, peer groups, and the enterprise. Support must be provided to allow for intelligent agents running on the edge to participate in guaranteed message delivery with enterprise transaction managers such as those provided in .NET's MTS, Java's Transaction Service, and OMG's Object Transaction Service.

## 9. Artificial/Cognitive Intelligence

Next-generation applications will need to utilize intelligent software agents that can gather data, respond quickly based on this data as it changes, produce and distribute knowledge, and possibly initiate other agent activities. The underlying rules engine must be easy to use, provide very high performance against potentially large rule sets, and must be available in .NET and Java.

## 10. Embedded and Enterprise Database Integration and Synchronization

Mobile 2.0 applications must provide a simple way to access databases, regardless of the type of database, be it relational, object, xml, and a multi-user enterprise database or single-user embedded. Developers need to be shielded from the intricacies that exist with these various flavors and have support for data synchronization between the edge and enterprise.

After citing a handful of examples, it is easy to see where Mobile 2.0 applications will have a huge impact from the social and entertainment life of the average person, to their health and safety as it relates to emergency response, disaster recovery, terrorist response, homeland security and military operations.

## The Mobile 2.0 Super Highway

By providing many of these capabilities today, Voyager Edge provides the superhighway for software engineers to leverage in creating the next generation of applications.

Software architects and engineers now have the flexibility to freely develop dynamic, intelligent, and decentralized applications in both .NET or Java, on the devices and servers they need to target, whether the device/node is a mainframe, desktop, PDA, Smartphone, RFID reader, camera, or embedded sensor.

By utilizing Voyager Edge, developers have at their disposal a choice of wireless networks, distributed protocols, and decentralized and centralized messaging capabilities to leverage in creating the ad-hoc and distributed social, knowledge, and problem-solving networks of tomorrow.

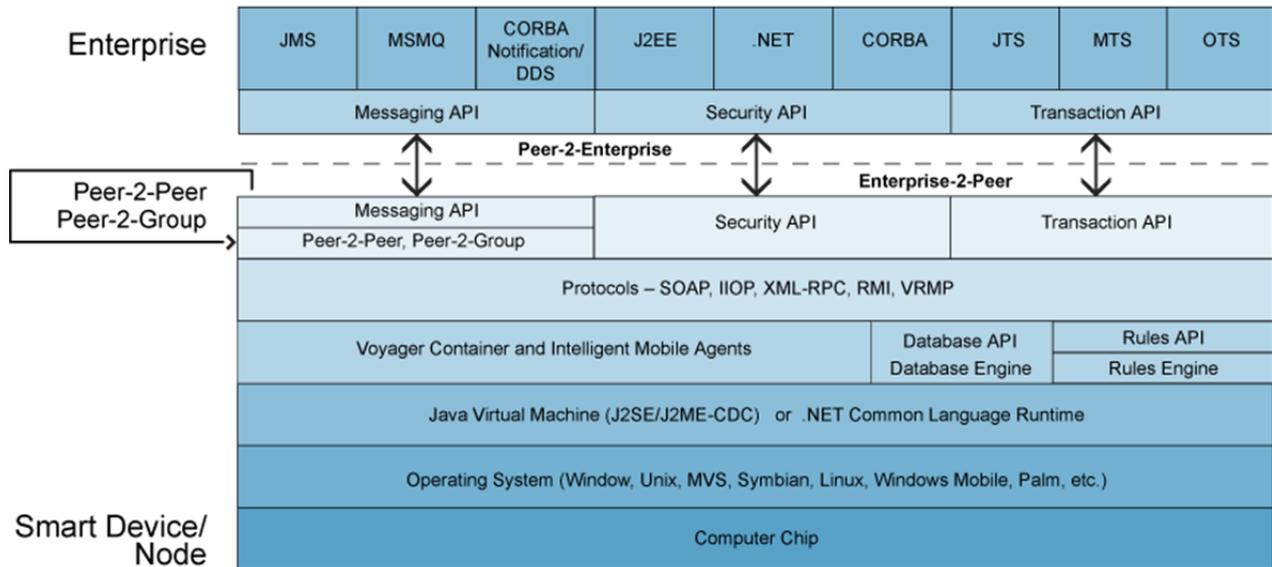
Utilizing the RETE-based rules engine integrated within Voyager Edge, engineers can create artificial intelligence and cognitive capabilities into Mobile 2.0 software agents residing on all targeted devices and servers. Engineers will be able to produce applications that provide real-time intelligence, situational awareness, and coordination at the edge not found today.

Enterprise software stacks, like software languages, continue to be born, but "old" ones never go away. J2EE and .NET architectures and related languages dominate, but CORBA and Mainframe systems and their associated languages, are still prevalent within many of these same organizations.

A very similar situation exists in the device world: where Windows Mobile, Symbian, Embedded Linux, Blackberry, PalmOS, and most recently Sun's JavaFX, compete with many other embedded operating systems. The same is true for wireless networks, some of which are mentioned above.

Recursion Software's Voyager Edge provides a single cohesive and comprehensive Mobile 2.0 platform that allows next-generation applications to seamlessly exist in this diverse software and device environment, and the increasingly varied wired and wireless networks used to connect them.

### Voyager Edge Software Stack



#### Sample Code

Believing that these Mobile 2.0 apps can be built today may require creating real-world example. To demonstrate Voyager Edge's capabilities, download a free [community edition of Voyager Edge](#). It comes with extensive documentation and over 40 examples covering all of the features of Voyager Edge, including examples in Java, C#, VB.NET and Managed C++.

#### 1. Creating, naming and moving a mobile object on/to any node

The following is a code snippet to:

1. create an object
2. bind it to our federated, decentralized Naming Service
3. move it

```
// create a remote component on the
local node
IStockmarket market1 =
(IStockmarket)
Factory.create("examples.stockmarket
.Stockmarket");

// create an remote component on
node "//dallas:8000"
IStockmarket market2 =
(IStockmarket)
Factory.create("examples.stockmarket
.Stockmarket", "//dallas:8000");

// bind to naming service
Namespace.bind("//dallas:8000/NASDAQ
", market );

// move to another node "//
tokyo:9000"
IMobility mobility = Mobility.of(
market ); // obtain mobility facet
3b.mobility.moveTo("//tokyo:9000" );
// move the object to a new location
```

## 2. Turning a remote component into an autonomous agent

The following is a code snippet to:

1. create an object
2. bind it to our federated, decentralized Naming Service
3. turns it into an agent
4. makes it autonomous
5. move it to another node

```
// create a remote trader component
on the local node
ITrader trader = (ITrader)
Factory.create(
Trader.class.getName() );

// create an Agent facet
IAgent aAgent =
AgentFacet.of(trader);

// setting to autonomous, prevents
it from being garbage collected
// default is true
aAgent.setAutonomous(true);

// move to node and reset it at
arrival by calling "atMarket"
method
aAgent.moveTo("//dallas:8000",
"atMarket" );
```

## 3. Creating and joining an object/agent community (a.k.a space) on any node

The following is a code snippet to:

1. create an agent space on a remote node
2. create a remote component and add it that that community
3. create another agent space on a remote node
4. create another remote component and add it that the community just created

```
// creating a space on a remote node
ISubspace subspace1 = (ISubspace)
Factory.create(
"com.recurionsw.ve.space.Subspace"
, "8000/Subspace1" );

// creating a remote component on
that node
```

```
IConsumer consumer1 = (IConsumer)
Factory.create(
"examples.space.Consumer", new
Object[]{ "jack" }, "8000/Jack" );

// adding the remote object to that
remote space
subspace1.add( consumer1 );

// creating another space on another
remote node
ISubspace subspace2 = (ISubspace)
Factory.create(
"com.recurionsw.ve.space.Subspace"
, "8000/Subspace1" );

// creating a remote component on
that node
IConsumer consumer2 = (IConsumer)
Factory.create(
"examples.space.Consumer", new
Object[]{ "jack" }, "8000/Jack" );

// adding the remote object to that
remote space
subspace2.add( consumer2 );
```

## 4. Joining two object/agent communities on any nodes

The following is a code snippet to:

1. connect or chain the two spaces that were previously created on different nodes

```
// connecting two agent communities
together
// regardless of what nodes they
are located on
subspace1.connect( subspace2 );
```

## 5. Multicasting a message within an object/agent community

The following is a code snippet to:

1. lookup an object/agent community
2. get a multicast proxy
3. call a method on any objects of type "consumer" located in that community

```
// looking up an agent community
ISubspace subspace1 = (ISubspace)
Namespace.lookup( "8000/Subspace1"
);
```

```
// get the multicast proxy
associated with Consumer agent
IConsumer consumer1 = (IConsumer)
subspace1.getMulticastProxy(

    "examples.space.IConsumer" );

// publish the message to the
community
// calls "news" method on every
Consumer object in the community
Multicast.invoke( subspace1,
"news", new Object[] { "newsflash
2!" },

    "examples.space.IConsumer" );
```

## 6. Exposing and accessing an object/agent via its automatically created Web Service

The following is a code snippet to:

1. enable web services support
2. call a method on a remote component using SOAP

```
// enable web services support
com.recurionsw.ve.web.services.Web
Services.enableWebServices();

// Create a service model for the
web service client
ServiceFactory serviceFactory = new
ObjectServiceFactory();
Service serviceModel =
serviceFactory.create(IStockmarket.c
lass);

// Create a client proxy
XFireProxyFactory proxyFactory = new
XFireProxyFactory();

String hostName =
InetAddress.getLocalHost().getHostNa
me();
IStockmarket market =
    (IStockmarket)proxyFactory.cre
ate(serviceModel,
    "http://dallas.recurionsw.com
:8000/services/Stockmarket");

// call "getQuote" using on
IstockMarket Web Service using SOAP
market.getQuote("sun");
```

## Creating a Mobile 2.0 World Today

The impact of Mobile 2.0 applications in the commercial world will as also be profoundly felt in the telecom, healthcare, energy, transportation, insurance, education, and finance industries. When one really stops to think about how far reaching this technology is, indeed there is not a vertical in the commercial, government or consumer arena that will not greatly benefit from the revolutionary capabilities this platform brings to the Mobile 2.0 world.

Some industry insiders predict that Mobile 2.0 apps remain a goal to be obtained in the next few years. Yet a platform exists today that can fulfill the vast majority of Mobile 2.0 requirements, with the remaining functionality to be delivered in the coming months. For the creative software engineer and architect, today is a golden opportunity to not only envision the killer new application for your organization, but to use Recursion Software's Voyager Edge, and your professional design and implementation skills to make your vision a reality. ■

---

For more information and white papers on mobile agent technology, visit:  
[recurionsw.com/Products/voyager.html](http://recurionsw.com/Products/voyager.html)

A free community version of Voyager Edge is available on the Recursion website.



### **About Recursion Software, Inc.**

Recursion Software is an innovative provider of intelligent middleware and distributed computing solutions based on Service Oriented Architecture (SOA) principles and interoperability standards between multiple languages and platforms. Recursion products help enterprises to extend their current application architecture while providing the tools developers need to build the next generation of intelligent, mobile, applications. The company is a small, privately held corporation, located in the Dallas-Fort Worth area with a large customer base of government and commercial clients across the world. Since 1993, our products have enabled complex, performance-oriented software development solutions for mission-

critical applications and systems. The majority of clients are in the defense, financial, energy, computer technology, and telecommunications industries.

Recursion Software is regarded for its Voyager Edge platform, a powerful agent-based interoperable platform that supports a total range of edge devices, including handheld devices, PDAs, sensors and cameras. The company remains the leading proponent and preferred platform for intelligent mobile agent and agent space technology and has been issued more than 18 patents related to distributed computing, with 30 patents in various states of pending and filing.

© 2007 Recursion Software, Inc.  
2591 North Dallas Parkway  
Suite 200  
Frisco, TX 75034

972.731.8800  
800.727.8674

[www.recursionsw.com](http://www.recursionsw.com)