

The Most Secure Graph Database Available

Triples offer a way of describing model elements and relationships between them. In some cases, however, it is also convenient to be able to store data that is associated with a triple as a whole rather than with a particular element. For instance one might wish to record the source from which a triple has been imported or access level necessary to include it in query results. Traditional solutions of this problem include using graphs, RDF reification or triple IDs. All of these approaches suffer from various flexibility and performance issues. For this reason AllegroGraph offers an alternative: triple attributes.

Attributes are key-value pairs associated with a triple. Keys refer to attribute definitions that must be added to the store before they are used. Values are strings. The set of legal values of an attribute can be constrained by the definition of that attribute. It is possible to associate multiple values of a given attribute with a single triple.

Possible uses for triple attributes include:

- *Access control: It is possible to instruct AllegroGraph to prevent an user from accessing triples with certain attributes.*
- *Sharding: Attributes can be used to ensure that related triples are always placed in the same shard when AllegroGraph acts as a distributed triple store.*

Like all other triple components, attribute values are immutable. They must be provided when the triple is added to the store and cannot be changed or removed later.

To illustrate the use of triple attributes we will construct an artificial data set containing a log of information about contacts detected by a submarine at a single moment in time.

Managing attribute definitions

Before we can add triples with attributes to the store we must create appropriate attribute definitions.

First let's open a connection

```
from franz.openrdf.connect import ag_connect

conn = ag_connect('python-tutorial', create=True, clear=True)
```

Attribute definitions are represented by **AttributeDefinition** objects. Each definition has a name, which must be unique, and a few optional properties (that can also be passed as constructor arguments):

- *allowed_values*: a list of strings. If this property is set then only the values from this list can be used for the defined attribute.
- *ordered*: a boolean. If true then attribute value comparisons will use the ordering defined by *allowed_values*. The default is false.
- *minimum_number*, *maximum_number*: integers that can be used to constrain the cardinality of an attribute. By default there are no limits.

Let's define a few attributes that we will later use to demonstrate various attribute-related capabilities of AllegroGraph. To do this, we will use the **setAttributeDefinition()** method of the connection object.

```
from franz.openrdf.repository.attributes import AttributeDefinition

# A simple attribute with no constraints governing the set
# of legal values or the number of values that can be
# associated with a triple.
tag = AttributeDefinition(name='tag')

# An attribute with a limited set of legal values.
# Every bit of data can come from multiple sources.
# We encode this information in triple attributes,
# since it refers to the tripe as a whole. Another
# way of achieving this would be to use triple ids
# or RDF reification.
source = AttributeDefinition(
    name='source',
    allowed_values=['sonar', 'radar', 'esm', 'visual'])

# Security level - notice that the values are ordered
# and each triple must have exactly one value for
# this attribute. We will use this to prevent some
# users from accessing classified data.
level = AttributeDefinition(
    name='level',
```

```

    allowed_values=['low', 'medium', 'high'],
    ordered=True,
    minimum_number=1,
    maximum_number=1)

# An attribute like this could be used for sharding.
# That would ensure that data related to a particular
# contact is never partitioned across multiple shards.
# Note that this attribute is required, since without
# it an attribute-sharded triple store would not know
# what to do with a triple.
contact = AttributeDefinition(
    name='contact',
    minimum_number=1,
    maximum_number=1)

# So far we have created definition objects, but we
# have not yet sent those definitions to the server.
# Let's do this now.
conn.setAttributeDefinition(tag)
conn.setAttributeDefinition(source)
conn.setAttributeDefinition(level)
conn.setAttributeDefinition(contact)

# This line is not strictly necessary, because our
# connection operates in autocommit mode.
# However, it is important to note that attribute
# definitions have to be committed before they can
# be used by other sessions.
conn.commit()

```

It is possible to retrieve the list of attribute definitions from a repository by using the **getAttributeDefinitions()** method:

```

for attr in conn.getAttributeDefinitions():
    print('Name: {0}'.format(attr.name))
    if attr.allowed_values:
        print('Allowed values: {0}'.format(
            ', '.join(attr.allowed_values)))
        print('Ordered: {0}'.format(
            'Y' if attr.ordered else 'N'))
    print('Min count: {0}'.format(attr.minimum_number))
    print('Max count: {0}'.format(attr.maximum_number))
    print()

```

Notice that in cases where the maximum cardinality has not been explicitly defined, the server replaced it with a default value. In practice this value is high enough to be interpreted as ‘no

limit'.

```
Name: tag
Min count: 0
Max count: 1152921504606846975

Name: source
Allowed values: sonar, radar, esm, visual
Min count: 0
Max count: 1152921504606846975
Ordered: N

Name: level
Allowed values: low, medium, high
Ordered: Y
Min count: 1
Max count: 1

Name: contact
Min count: 1
Max count: 1
```

Attribute definitions can be removed (provided that the attribute is not used by the static attribute filter, which will be discussed later) by calling **deleteAttributeDefinition()**:

```
conn.deleteAttributeDefinition('tag')
defs = conn.getAttributeDefinitions()
print(', '.join(sorted(a.name for a in defs)))
```

```
contact, level, source
```

Adding triples with attributes

Now that the attribute definitions have been established we can demonstrate the process of adding triples with attributes. This can be achieved using various methods. A common element of all these methods is the way in which triple attributes are represented. In all cases dictionaries with attribute names as keys and strings or lists of strings as values are used.

When **addTriple()** is used it is possible to pass attributes in a keyword parameter, as shown below:

```
ex = conn.namespace('ex://')
```

```
conn.addTriple(ex.S1, ex.cls, ex.Udaloy, attributes={
    'source': 'sonar',
    'level': 'low',
    'contact': 'S1'
})
```

The **addStatement()** method works in similar way. Note that it is not possible to include attributes in the **Statement** object itself.

```
from franz.openrdf.model import Statement

s = Statement(ex.M1, ex.cls, ex.Zumwalt)
conn.addStatement(s, attributes={
    'source': ['sonar', 'esm'],
    'level': 'medium',
    'contact': 'M1'
})
```

When adding multiple triples with **addTriples()** one can add a fifth element to each tuple to represent attributes. Let us illustrate this by adding an aircraft to our dataset.

```
conn.addTriples(
    [(ex.R1, ex.cls, ex['Ka-27'], None,
      {'source': 'radar',
       'level': 'low',
       'contact': 'R1'}),
     (ex.R1, ex.altitude, 200, None,
      {'source': 'radar',
       'level': 'medium',
       'contact': 'R1'})])
```

When all or most of the added triples share the same attribute set it might be convenient to use the **attributes** keyword parameter. This provides default values, but is completely ignored for all tuples that already contain attributes (the dictionaries are not merged). In the example below we add a triple representing an aircraft carrier and a few more triples that specify its position. Notice that the first triple has a lower security level and multiple sources. The common ‘contact’ attribute could be used to ensure that all this data will remain on a single shard.

```
conn.addTriples(
    [(ex.M2, ex.cls, ex.Kuznetsov, None, {
        'source': ['sonar', 'radar', 'visual'],
        'contact': 'M2',
```

```

        'level': 'low',
    )),
    (ex.M2, ex.position, ex.pos343),
    (ex.pos343, ex.x, 430.0),
    (ex.pos343, ex.y, 240.0)],
    attributes={
        'contact': 'M2',
        'source': 'radar',
        'level': 'medium'
    })

```

Another method of adding triples with attributes is to use the NQX file format. This works both with **addFile()** and **addData()** (illustrated below):

```

from franz.openrdf.rio.rdfformat import RDFFormat

conn.addData('''
    <ex://S2> <ex://cls> <ex://Alpha> \N
    {"source": "sonar", "level": "medium", "contact": "S2"} .
    <ex://S2> <ex://depth> "300" \N
    {"source": "sonar", "level": "medium", "contact": "S2"} .
    <ex://S2> <ex://speed_kn> "15.0" \N
    {"source": "sonar", "level": "medium", "contact": "S2"} .
''', rdf_format=RDFFormat.NQX)

```

When importing from a format that does not support attributes, it is possible to provide a common set of attribute values with a keyword parameter:

```

from franz.openrdf.rio.rdfformat import RDFFormat

conn.addData('''
    <ex://V1> <ex://cls> <ex://Walrus> ;
        <ex://altitude> 100 ;
        <ex://speed_kn> 12.0e+8 .
    <ex://V2> <ex://cls> <ex://Walrus> ;
        <ex://altitude> 200 ;
        <ex://speed_kn> 12.0e+8 .
    <ex://V3> <ex://cls> <ex://Walrus> ;
        <ex://altitude> 300;
        <ex://speed_kn> 12.0e+8 .
    <ex://V4> <ex://cls> <ex://Walrus> ;
        <ex://altitude> 400 ;
        <ex://speed_kn> 12.0e+8 .
    <ex://V5> <ex://cls> <ex://Walrus> ;
        <ex://altitude> 500 ;
        <ex://speed_kn> 12.0e+8 .
''')

```

```

        <ex://V6> <ex://cls> <ex://Walrus> ;
            <ex://altitude> 600 ;
            <ex://speed_kn> 12.0e+8 .
''' , attributes={
    'source': 'visual',
    'level': 'high',
    'contact': 'a therapist'})

```

The data above represents six visually observed Walrus-class submarines, flying at different altitudes and well above the speed of light. It has been highly classified to conceal the fact that someone has clearly been drinking while on duty – after all there are only four Walrus-class submarines currently in service, so the observation is obviously incorrect.

Retrieving attribute values

We will now print all the data we have added to the store, including attributes, to verify that everything worked as expected. The only way to do that is through a SPARQL query using the appropriate [magic property](#) to access the attributes. The query below binds a literal containing a JSON representation of triple attributes to the `?a` variable:

```

import json

r = conn.executeTupleQuery('''
    PREFIX attr: <http://franz.com/ns/allegrograph/6.2.0/>
    SELECT ?s ?p ?o ?a {
        ?s ?p ?o .
        ?a attr:attributes (?s ?p ?o) .
    } ORDER BY ?s ?p ?o''')
with r:
    for row in r:
        print(row['s'], row['p'], row['o'])
        print(json.dumps(json.loads(row['a'].label),
                          sort_keys=True,
                          indent=4))

```

The result contains all the expected triples with pretty-printed attributes.

```

<ex://M1> <ex://cls> <ex://Zumwalt>
{
    "contact": "M1",
    "level": "medium",

```

```
        "source": [
            "esm",
            "sonar"
        ]
    }
    <ex://M2> <ex://cls> <ex://Kuznetsov>
    {
        "contact": "M2",
        "level": "low",
        "source": [
            "visual",
            "radar",
            "sonar"
        ]
    }
    <ex://M2> <ex://position> <ex://pos343>
    {
        "contact": "M2",
        "level": "medium",
        "source": "radar"
    }
    <ex://R1> <ex://altitude> "200"^^...
    {
        "contact": "R1",
        "level": "medium",
        "source": "radar"
    }
    <ex://R1> <ex://cls> <ex://Ka-27>
    {
        "contact": "R1",
        "level": "low",
        "source": "radar"
    }
    <ex://S1> <ex://cls> <ex://Udaloy>
    {
        "contact": "S1",
        "level": "low",
        "source": "sonar"
    }
    <ex://S2> <ex://cls> <ex://Alpha>
    {
        "contact": "S2",
        "level": "medium",
        "source": "sonar"
    }
    <ex://S2> <ex://depth> "300"
    {
        "contact": "S2",
        "level": "medium",
        "source": "sonar"
    }
```

```

}
<ex://S2> <ex://speed_kn> "15.0"
{
  "contact": "S2",
  "level": "medium",
  "source": "sonar"
}
<ex://V1> <ex://altitude> "100"^^...
{
  "contact": "a therapist",
  "level": "high",
  "source": "visual"
}
<ex://V1> <ex://cls> <ex://Walrus>
{
  "contact": "a therapist",
  "level": "high",
  "source": "visual"
}
<ex://V1> <ex://speed_kn> "1.2E9"^^...
{
  "contact": "a therapist",
  "level": "high",
  "source": "visual"
}
...
<ex://pos343> <ex://x> "4.3E2"^^...
{
  "contact": "M2",
  "level": "medium",
  "source": "radar"
}
<ex://pos343> <ex://y> "2.4E2"^^...
{
  "contact": "M2",
  "level": "medium",
  "source": "radar"
}
}

```

Attribute filters

Triple attributes can be used to provide fine-grained access control. This can be achieved by using [static attribute filters](#).

Static attribute filters are simple expressions that control which triples are visible to a query based on triple attributes. Each repository has a single, global attribute filter that can be modified using **setAttributeFilter()**. The values passed to this

method must be either strings (the syntax is described in the documentation of [static attribute filters](#)) or filter objects. Filter objects are created by applying set operators to 'attribute sets'. These can then be combined using filter operators. An attribute set can be one of the following:

- *a string or a list of strings: represents a constant set of values.*
- *TripleAttribute.name: represents the value of the name attribute associated with the currently inspected triple.*
- *UserAttribute.name: represents the value of the name attribute associated with current query. User attributes will be discussed in more detail later.*

Available set operators are shown in the table below. All classes and functions mentioned here can be imported from the `franz.openrdf.repository.attributes` package:

Syntax	Meaning
<code>Empty(x)</code>	True if the specified attribute set is empty.
<code>Overlap(x, y)</code>	True if there is at least one matching value between the two attribute sets.
<code>Subset(x, y), x << y</code>	True if every element of x can be found in y
<code>Superset(x, y), x >> y</code>	True if every element of y can be found in x
<code>Equal(x, y), x == y</code>	True if x and y have exactly the same contents.
<code>Lt(x, y), x < y</code>	True if both sets are singletons, at least one of the sets refers to a triple or user attribute, the attribute is ordered and the value of the single element of x occurs before the single value of y in the <code>lowed_values</code> list of the attribute.

Syntax	Meaning
<code>Le(x, y), x <= y</code>	True if $y < x$ is false.
<code>Eq(x, y)</code>	True if both $x < y$ and $y < x$ are false. Note that using the <code>==</code> Python operator translates to <i>Eqauls</i> , not <i>Eq</i> .
<code>Ge(x, y), x >= y</code>	True if $x < y$ is false.
<code>Gt(x, y), x > y</code>	True if $y < x$.

Note that the overloaded operators only work if at least one of the attribute sets is a `UserAttribute` or `TripleAttribute` reference – if both arguments are strings or lists of strings the default Python semantics for each operator are used. The prefix syntax always produces filters.

Filters can be combined using the following operators:

Syntax	Meaning
<code>Not(x), ~x</code>	Negates the meaning of the filter.
<code>And(x, y, ...), x & y</code>	True if all subfilters are true.
<code>Or(x, y, ...), x y</code>	True if at least one subfilter is true.

Filter operators also work with raw strings, but overloaded operators will only be recognized if at least one argument is a filter object.

Using filters and user attributes

The example below displays all classes of vessels from the dataset after establishing a static attribute filter which ensures that only sonar contacts are visible:

```
from franz.openrdf.repository.attributes import *

conn.setAttributeFilter(TripleAttribute.source >> 'sonar')
conn.executeTupleQuery(
    'select ?class { ?s <ex://cls> ?class } order by ?class',
    output=True)
```

The output contains neither the visually observed Walruses nor the radar detected ASW helicopter.

```

-----
| class          |
=====
| ex://Alpha     |
| ex://Kuznetsov |
| ex://Udaloy    |
| ex://Zumwalt   |
-----

```

To avoid having to set a static filter before each query (which would be inefficient and cause concurrency issues) we can employ user attributes. User attributes are specific to a particular connection and are sent to the server with each query. The static attribute filter can refer to these and compare them with triple attributes. Thus we can use code presented below to create a filter which ensures that a connection only accesses data at or below the chosen clearance level.

```

conn.setUserAttributes({'level': 'low'})
conn.setAttributeFilter(
    TripleAttribute.level <= UserAttribute.level)
conn.executeTupleQuery(
    'select ?class { ?s <ex://cls> ?class } order by ?class',
    output=True)

```

We can see that the output here contains only contacts with the access level of *low*. It omits the destroyer and Alpha submarine (these require *medium* level) as well as the top-secret Walruses.

```

-----
| class          |
=====
| ex://Ka-27     |
| ex://Kuznetsov |
| ex://Udaloy    |
-----

```

The main advantage of the code presented above is that the filter can be set globally during the application setup and access control can then be achieved by varying user attributes on connection objects.

Let us now remove the attribute filter to prevent it from interfering with other examples. We will use the **clearAttributeFilter()** method.

```
conn.clearAttributeFilter()
```

It might be useful to change connection's attributes temporarily for the duration of a single code block and restore prior attributes after that. This can be achieved using the **temporaryUserAttributes()** method, which returns a context manager. The example below illustrates its use. It also shows how to use **getUserAttributes()** to inspect user attributes.

```
with conn.temporaryUserAttributes({'level': 'high'}):  
    print('User attributes inside the block:')  
    for k, v in conn.getUserAttributes().items():  
        print('{0}: {1}'.format(k, v))  
    print()  
print('User attributes outside the block:')  
for k, v in conn.getUserAttributes().items():  
    print('{0}: {1}'.format(k, v))
```

```
User attributes inside the block:  
level: high
```

```
User attributes outside the block:  
level: low »
```

The marvels of an event-based schema

Franz's CEO, Jans Aasman, recently wrote the following article for InfoWorld.



When working with various data types at the speed of big data, this method is ideal for integrating and aggregating

assorted information for the holistic value it provides.

The issue of schema—and what is frequently perceived as its inherent difficulties—is becoming more important every day. Organizations are increasingly encountering decentralized computing environments typified by semi-structured or unstructured external data of varying formats, often requiring integration with internal, structured data for immediate business value.

Read the Full Article

Enterprise Data Modeling Made Easy

From Analytics Week:

Enterprise data modeling has remained an arduous, time-consuming task for myriad reasons, not the least of which is the different levels of modeling required across an organization's various business domains.

Data modelers have to consider conceptual, logical and physical models, in addition to those for individual databases, applications, and a variety of environments such as production and post-production. Oftentimes, the need to integrate new sources or to adapt to changing business or technology requirements exacerbates this process, causing numerous aspects of it to essentially begin all over again.

Enterprise data modeling is rendered much more simply with the incorporation of semantic technologies—particularly when compared to traditional relational ones. Nearly all of the

foregoing modeling layers are simplified into an evolving semantic model that utilizes a standards-based approach to harmonize modeling concerns across an organization, its domains, and data environments.

Moreover, the semantic approach incorporates visual aspects that allows modelers to discern relationships between objects and readily identify them with a degree of precision that would require long periods of time with relational technologies.

“Semantics are designed for sharing data,” **Franz** CEO Jans Aasman reflected. “Semantic data flows into how people think.”

Read the full article:

Franz Inc. and The Wrocław Institute of Spatial Information and Artificial Intelligence (The Wrocław Institute) team up to deliver graph and A.I. solutions in Poland

A Wrocław Institute News Release

OAKLAND, Calif. – March 15, 2016 – We are pleased to inform that Wrocław Institute has been appointed as a partner by Franz Inc.— world’s leading producer of semantic graph

technologies. The agreement grants to Wroclaw Institute exclusive right to sell Franz's – AllegroGraph family of products for territory of Poland. AllegroGraph is best in class graph database, fully supporting W3C standards adopted by start-up's as well as vast number of Fortune 100 companies. AllegroGraph is a part of Big Data ecosystem as it could be integrated with Apache Hadoop and Amazon EC2.

The Wroclaw Institute CEO – Dr. Adam Iwaniak said "Partnership with Franz Inc. is a turning point for our company as semantic graph technology is gaining a lot of market attention in 'data tsunami' era. We are happy that we will be able to provide our customers with award winning solution to help them manage their complex data resources. Moreover I'd like to emphasize that as a company we made a big progress in leveraging RDF graphs technologies also on our basic market – geoinformatics".

"We are excited about the opportunity to work with Dr. Iwaniak and the Wroclaw Institute team to demonstrate why Graph Databases deliver new, real time decision making capabilities for the Enterprise." said Dr. Jans Aasman, CEO, Franz Inc., "Organizations across Poland will benefit from AllegroGraph's ability to link highly complex data, generating new knowledge and insight for a significant competitive advantage."

AllegroGraph is a database technology that enables businesses to extract sophisticated decision insights and predictive analytics from their highly complex, distributed data that can't be answered with conventional databases. Unlike traditional relational databases, Franz's product AllegroGraph employs a combination of semantic, graph and spatial technologies that process data with contextual and conceptual intelligence. AllegroGraph is able to run queries of unprecedented complexity to support predictive analytics that help companies make better, real-time decisions.

AllegroGraph is commonly used in defense and intelligence,

banking, and insurance, pharmaceutical, and healthcare, Linked Data publishing, as well as by organization dealing with complex, constantly changing knowledge bases.

About Franz Inc.

Franz Inc. is a leading vendor of semantic technology tools featuring AllegroGraph – high-performance, scalable, disk-based graph database, provides the solid storage layer for powerful GeoTemporal Reasoning, Social Network Analytics and Ontology Modeling. Based in Oakland, California, Franz Inc. is an American owned company that delivers leading-edge development products that enable software developers to build flexible, scalable, semantic applications quickly and cost-effectively.

About The Wroclaw Institute

The Wroclaw Institute of Spatial Information and Artificial Intelligence is Wroclaw, Poland based technology company focused on knowledge engineering, data exploration and intelligent GIS providing products, services and solutions based on cutting-edge scientific and technological achievements.

Related Links

- WIZIPISI dystrybutorem oprogramowania AllegroGraph
- Oprogramowanie bazodanowe AllegroGraph dostepne w Polsce
- Wroclaw Institute of Spatial Information and Artificial Intelligence

All trademarks and registered trademarks in this document are the properties of their respective owners.

AllegroGraph Recognized as Best in Semantic Web Technology – USA & Leader in Graph Database Products

Franz's AllegroGraph Fueling Rapid Growth in Graph Database Category

OAKLAND, Calif. – February 3, 2016 – Franz Inc., an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology has been recognized As “Best in Semantic Web Technology – USA & Leader in Graph Database Products” by Corporate America Software and Technology.

“At Corporate America, it’s our priority to showcase prominent professionals who are excelling in their industry and outperforming their competitors,” said Hannah Stevenson, Managing Group Editor, AI Global Media. “Franz Inc. have a reputation for innovation, utilizing their expert knowledge to create complex and exciting Graph Database solutions. Franz’s unique platforms offer highly scalable technologies for solving complex Big Data challenges.”

Corporate America is the definitive magazine for CEOs, top tier management and key decision makers across the US. Created to inform, influence, and shape the corporate conversation across the nation through high quality editorial, in-depth research and an experienced and dedicated network of advisers, Corporate America provides its readership with the most authoritative and current analysis of the major changes effecting the corporate landscape, and the latest deals and

topical issues dominating the corporate universe. A multifaceted program, the awards are focused on rewarding excellence across all areas of the technology and software industries and all nominees are closely scrutinized to ensure that only the most deserving receive Corporate America's prestigious awards.

"We are excited that Graph Databases, like AllegroGraph, have garnered the attention they deserve by Enterprise customers looking to innovate," said Dr. Jans Aasman, CEO, Franz Inc. "In today's data-driven environments, the ability to quickly analyze data from diverse sources is becoming critical. We are already seeing how Semantic Graph Databases with predictive analytics can help transform healthcare through Precision Medicine and make us safer through Insider Threat Detection."

"Because it (AllegroGraph) is a Graph database, it can store pretty much any kind of data and query it, not just in the time-worn relational fashion, but also in a graphical manner – carving out graphical maps of relationships. And on top of that, it can apply semantics to deduce as-yet-undiscovered knowledge from the data. Its capabilities are very broad, and they provide a glimpse of the shape of things to come," added Bloor. stated Robin Bloor, co-founder and Chief Analyst of The Bloor Group.

"Information has always existed everywhere but has often been isolated, incomplete, unavailable or unintelligible," according to Gartner. "Advances in semantic tools such as graph databases as well as other emerging data classification and information analysis techniques will bring meaning to the often chaotic deluge of information." (Source: Gartner Identifies the Top Strategic Technology Trends for 2016.)

A recent Forrester Research report stated, "Graph databases are a powerful optimized technology that link billions of pieces of connected data to help create new sources of value for customers and increase operational agility for customer

service. Because graph databases track connections among entities and offer links to get more detailed information, they are well-suited for scenarios in which relationships are important, such as cybersecurity, social network analysis, eCommerce recommendations, dependence analysis, and predictive analytics.” (Source: Forrester Research, Market Overview: Graph Databases, May 28, 2015)

Franz’s recent announcement of the first Semantic Data Lake (SDL) for Healthcare, which was created in collaboration with Montefiore Medical Center (the eighth largest hospital group in the U.S.), Intel, Cloudera and Cisco. The SDL for Healthcare is a scalable and extensible Healthcare platform designed for Accountable Care and Personalized Medicine initiatives. AllegroGraph has played a critical role in the Semantic Data Lake for Healthcare, by facilitating integration of complex information for basic science, clinical, population, community, environmental, behavioral and wellness research data to enable knowledge-based analytics, classification, pattern recognition, predictive modeling and simulations at scale.

About Corporate America

Corporate America is more than just a magazine. Alongside our quarterly publication, we also produce a website that is regularly updated with the latest news, features, opinion and comment, again in conjunction with a host of top-level advisers, experts and businesspeople, and throughout the year, you’ll also get your chance to participate in our highly regarded awards programs, designed to pay tribute to the finest firms and individuals on the American business landscape.

About AllegroGraph

Unlike traditional relational databases or Property Graph Databases, AllegroGraph employs semantic graph technologies

that process data with contextual and conceptual intelligence. AllegroGraph is able run queries of unprecedented complexity to support predictive analytics that help organizations make more informed, real-time decisions. AllegroGraph is the first Graph Database to support analysis across N-dimensions – any conceivable measurement of an object, property or operation. AllegroGraph can analyze temporal (time) and geospatial (location) dimensions relative to any ‘event,’ such as a disease, drug interaction, genetic combination, biomarkers, observations, image or physical sensors. AllegroGraph is utilized by dozens of the top Fortune 500 companies worldwide.

About Franz Inc.

Franz Inc. is an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology with expert knowledge in developing and deploying complex Big Data analytics solutions. AllegroGraph, Franz’s flagship, high-performance, transactional, and scalable Semantic Graph Database, provides the solid storage layer for Enterprise grade NoSQL solutions. AllegroGraph’s Activity Recognition capabilities provides a powerful means to aggregate and analyze data about individual and organizational behaviors, preferences, relationships, plus spatial and temporal linkages between individuals and groups. For additional Franz Inc customer success stories please visit:

- AllegroGraph – <http://franz.com/agraph/success/>
- Allegro CL – <http://franz.com/success/>

Franz’s Professional Service team is in the business of helping companies turn Data into Information and Information into Knowledge. We combine Data, Business Intelligence, and Analytics consulting services under one roof for our customers. Franz, an American owned company based in Oakland, California, is committed to market-driven product development, the highest levels of product quality and responsive customer support and service. Franz customers include dozens of Fortune

500 companies and span the healthcare, government, life sciences and telecommunications industries worldwide. Franz has demonstrated consistent growth and profitability since inception.

All trademarks and registered trademarks in this document are the properties of their respective owners.

AllegroGraph Recognized Among Top 10 Analytics Solution Providers by Pharma Tech Outlook

Franz's AllegroGraph powers Pharma Analytics for Sophisticated Decision Insights from Complex, Distributed Big Data

OAKLAND, Calif. – January 28, 2016 – Franz Inc., an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology has been named to Pharma Tech Outlook's Top 10 Analytics Solutions Providers for 2016.

"Franz Inc. has been selected as a Top 10 Analytics Solution Provider after careful evaluation across a dozen quantitative and qualitative elements," said Stacey Smith, Editor of Pharma Tech Outlook. "Our selection process takes into consideration a company's experience, industry recognition, technical certifications, market presence and positive client reviews. Franz Inc. and their Semantic Graph Database, AllegroGraph, are clear market leaders for Analytics in the Pharmaceutical Industry."

Pharma Tech Outlook covers the latest developments in the pharmaceutical industry. They provide valuable updates – news, views and trends, expert opinions, studies, discoveries, R&D and clinical trials – essential for decision-makers in the industry. Covering all the novel outcomes, Pharma Tech Outlook aims at contributing to the transformation of innovations into services as well as creating a healthy and productive society.

Pharma Tech Outlook's "Top 10 Analytics Solution Providers" are selected annually by a panel of experts and members of Pharma Tech Outlook's editorial board to recognize and promote technology entrepreneurship.

"Using AllegroGraph, Enterprises can run queries of unprecedented complexity to enable predictive analytics and real time decision-making within a myriad of industries including Healthcare, Life Sciences, Financial Services, and Publishing," said Jans Aasman, CEO of Franz Inc. "Integrating databases is a virtually effortless which is particularly valuable if organizations want to tap into the growing number of public datasets to enrich their analytics."

"Information has always existed everywhere but has often been isolated, incomplete, unavailable or unintelligible," according to Gartner. "Advances in semantic tools such as graph databases as well as other emerging data classification and information analysis techniques will bring meaning to the often chaotic deluge of information." (Source: Gartner Identifies the Top Strategic Technology Trends for 2016.)

A recent Forrester Research report stated, "Graph databases are a powerful optimized technology that link billions of pieces of connected data to help create new sources of value for customers and increase operational agility for customer service. Because graph databases track connections among entities and offer links to get more detailed information, they are well-suited for scenarios in which relationships are important, such as cybersecurity, social network analysis,

eCommerce recommendations, dependence analysis, and predictive analytics.” (Source: Forrester Research, Market Overview: Graph Databases, May 28, 2015)

Franz’s recent announcement of the first Semantic Data Lake (SDL) for Healthcare, which was created in collaboration with Montefiore Medical Center (the eighth largest hospital group in the U.S.), Intel, Cloudera and Cisco. The SDL for Healthcare is a scalable and extensible Healthcare platform designed for Accountable Care and Personalized Medicine initiatives. AllegroGraph has played a critical role in the Semantic Data Lake for Healthcare, by facilitating integration of complex information for basic science, clinical, population, community, environmental, behavioral and wellness research data to enable knowledge-based analytics, classification, pattern recognition, predictive modeling and simulations at scale.

About Pharma Tech Outlook

Pharma Tech Outlook is an online and a monthly magazine which covers most important and latest developments in the pharmaceutical industry. Through nominations and consultations with industry leaders, its editors choose the best in Pharma domains. Pharma Tech Outlook’s December-January Edition is an annual listing of Top 10 Analytics Solution Providers. For more information, visit the website at: <http://www.pharmatechoutlook.com/>

About AllegroGraph

Unlike traditional relational databases or Property Graph Databases, AllegroGraph employs semantic graph technologies that process data with contextual and conceptual intelligence. AllegroGraph is able run queries of unprecedented complexity to support predictive analytics that help organizations make more informed, real-time decisions. AllegroGraph is the first Graph Database to support analysis across N-dimensions – any

conceivable measurement of an object, property or operation. AllegroGraph can analyze temporal (time) and geospatial (location) dimensions relative to any 'event,' such as a disease, drug interaction, genetic combination, biomarkers, observations, image or physical sensors. AllegroGraph is utilized by dozens of the top Fortune 500 companies worldwide.

About Franz Inc.

Franz Inc. is an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology with expert knowledge in developing and deploying complex Big Data analytics solutions. AllegroGraph, Franz's flagship, high-performance, transactional, and scalable Semantic Graph Database, provides the solid storage layer for Enterprise grade NoSQL solutions. AllegroGraph's Activity Recognition capabilities provides a powerful means to aggregate and analyze data about individual and organizational behaviors, preferences, relationships, plus spatial and temporal linkages between individuals and groups. For additional Franz Inc customer success stories please visit:

- AllegroGraph – <http://franz.com/agraph/success/>
- Allegro CL – <http://franz.com/success/>

Franz's Professional Service team is in the business of helping companies turn Data into Information and Information into Knowledge. We combine Data, Business Intelligence, and Analytics consulting services under one roof for our customers. Franz, an American owned company based in Oakland, California, is committed to market-driven product development, the highest levels of product quality and responsive customer support and service. Franz customers include dozens of Fortune 500 companies and span the healthcare, government, life sciences and telecommunications industries worldwide. Franz has demonstrated consistent growth and profitability since inception.

All trademarks and registered trademarks in this document are the properties of their respective owners.

Franz's Gruff Produces Dynamic Visual Discovery for Graph Analytics

Gruff and AllegroGraph Power Visual Graph Search and Visual Query Building for Banking, Healthcare, Pharma Discovery and Security Applications

OAKLAND, Calif. – November 2, 2015 – Franz Inc., an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology, today announced Gruff v6.0, the industry's leading Graph Visualization software for exploring and discovering connections within data. Gruff provides novice users and graph experts the ability to visually build queries and visualize connections between data without writing code, which speeds discovery and enhances the ability to uncover hidden connections within data.

"Gruff allows for easy viewing of graph style data and provides an easy on-ramp for non-technical users to explore connections in their data," said Dr. Jans Aasman, CEO of Franz Inc. "Users can easily create queries visually, without becoming a query language expert, which further empowers the business user for this technology. Power users also benefit by creating ever more detailed queries in order extract knowledge from their data."

Gruff v6.0 produces dynamic data visualizations that organize connections between data in views that are driven by the user.

This visual flexibility can instantly unveil new discoveries and knowledge that turn complex data into actionable business insights. Gruff was developed by Franz to address Graph Search in large data sets and empower users to intelligently explore graphs in multiple views including:

- Graphical View – See the shape and density of graph data
- Tabular view – Understand objects as a whole
- Outline view – Explore the often hierarchical nature of graphs
- Query view – Write Prolog or SPARQL queries
- Graphical Query Builder – Create queries visually via drag and drop

Franz was recently named by CIOReview as one of the Top 20 Most Promising Database Solutions providers, due in part to the unique discovery capabilities offered by the combination of Gruff and AllegroGraph, Franz's Semantic Graph Database technology.

“Franz has been leading the burgeoning Graph Database revolution with a highly sophisticated, yet elegant Semantic Graph database solution,” said Harvi Sachar, Publisher & Founder, CIO Review. “Franz's AllegroGraph continues to break new ground in predictive analytics and visual graph discovery capabilities- benefiting organizations around the globe within Healthcare, Intelligence/National Security, Life Sciences and Financial Services.”

The popularity of Graph databases has skyrocketed – growing nearly 500% in the past two years, according to a ranking by DB-Engines. One reason for this growth is interest in using graph databases, rather than relational databases, to store master data. Graph databases offer a 360-degree view of master data and can answer questions about data relationships in real time, providing new, actionable insights from existing data.

A recent Dataversity article by Jelani Harper noted, “There is

a considerable degree of complexity in MDM systems in a business climate impacted by Big Data, especially for systems centered on customer domains. Numerous external sources (including social media and various forms of sentiment analyses) considerably complicate key relationships for products and customers. The deployment of graph databases, such as Franz's AllegroGraph, with MDM can simplify these relationships by visually representing the way that different categories of an organization's core business—based on ontologies—relate to one another.”

Gruff and AllegroGraph also play a pivotal role in the Semantic Data Lake for Healthcare. A collaboration with Franz Inc., Montefiore Medical Center (the eighth largest hospital in the U.S.), Intel, Cloudera and Cisco, to provide a scalable and extensible Healthcare platform designed for Accountable Care and Personalized Medicine initiatives.

“Making sense out of big data is a challenge, particularly in the healthcare industry where information comes from a variety of sources and in different forms including structured, unstructured, images, temporal, geo-location and signal data,” said Dr. Aasman, “With Gruff as part of the Semantic Data Lake platform, we can perform visual data exploration to discover new relationships between data that can save lives and improve care.”

Franz Inc. will host a Webcast on November 18th at 10AM PST, “Enriching the Property Graph with Relationship Objects,” which will demonstrate the power of Gruff and AllegroGraph for an online banking application, a fraud detection application for a European tax office, a machine learning application in healthcare and the CrunchBase investment database.

Gruff 6.0 Availability

Gruff 6.0 is available as a free download from the AllegroGraph website. The product runs on Mac OSX, Windows,

Linux and is offered as a standalone application or client-server for remote users.

About AllegroGraph

Unlike traditional relational databases or Property Graph Databases, AllegroGraph employs semantic graph technologies that process data with contextual and conceptual intelligence. AllegroGraph is able run queries of unprecedented complexity to support predictive analytics that help organizations make more informed, real-time decisions. AllegroGraph is the first Graph Database to support analysis across N-dimensions – any conceivable measurement of an object, property or operation. For example, AllegroGraph can analyze temporal (time) and geospatial (location) dimensions relative to any ‘event,’ such as a disease, drug interaction, genetic combination, biomarkers, observations, image or physical sensors.

About Franz Inc.

Franz Inc. is an innovative technology company with expert knowledge in developing and deploying Graph Search solutions. AllegroGraph, Franz’s flagship, high-performance, transactional, and scalable Graph Database, provides the solid storage layer for powerful Enterprise grade NoSQL solutions. AllegroGraph’s Activity Recognition capabilities provides a powerful means to aggregate and analyze data about individual and organizational behaviors, preferences, relationships, plus spatial and temporal linkages between individuals and groups. For additional Franz Inc customer success stories please visit:

- AllegroGraph – <http://franz.com/agraph/success/>
- Allegro CL – <http://franz.com/success/>

Franz’s Professional Service team is in the business of helping companies turn Data into Information and Information

into Knowledge. We combine Data, Business Intelligence, and Analytics consulting services under one roof for our customers. Franz, an American owned company based in Oakland, California, is committed to market-driven product development, the highest levels of product quality and responsive customer support and service. Franz customers include Fortune 500 companies in the government, life sciences and telecommunications industries. Franz has demonstrated consistent growth and profitability since inception.

All trademarks and registered trademarks in this document are the properties of their respective owners.

Bloor Research Positions AllegroGraph as a ‘Champion’ in Burgeoning Graph Database Market

Graph Databases Identified as the Fastest Growing Segment of the Database Market

OAKLAND, Calif. – May 5, 2015 – Franz, Inc., the leading supplier of Semantic Graph Database technology, today announced its flagship product, AllegroGraph, has been named a Champion by Bloor Research in its recent Graph Database Market Update report. AllegroGraph is a high performance Semantic Graph Database that enables analytics by leveraging the W3C industry standards. **Graph databases are skyrocketing in popularity and** have grown by 400% in the past two years, according to a recent DBMS ranking by DB-Engines.

“We are excited that Graph and RDF Databases are beginning to get the attention that they deserve,” said Dr. Jans Aasman, CEO, Franz Inc. “In today’s data-driven environments, the ability to quickly analyze data from diverse sources is becoming critical. We are already seeing how Semantic Graph Databases with predictive analytics can help transform healthcare through Precision Medicine and make us safer through Insider Threat Detection.”

“Graph databases handle a class of issues that are too structured for NoSQL and too diverse for relational technologies,” according to Bloor Research. “Relational databases are inherently limited to one-to-one, many-to-one and one-to-many relationships. They do not cater well for problems (such as bill of materials – a classic case) that are many-to-many. For these types of requirements graph databases not only perform way better than relational databases, but they allow some types of query that are simply not possible otherwise. Semantic query support tends to be particularly strong in triple stores. Another major point is that research suggests that graph visualizations are very easy and intuitive for users.” (Source: Bloor Research, Graph Databases, Philip Howard, April 13, 2015)

About AllegroGraph

AllegroGraph is a database technology that enables businesses to extract sophisticated decision insights and predictive analytics from highly complex, distributed data that cannot be uncovered with conventional databases. Unlike traditional databases or NoSQL databases, AllegroGraph employs semantic graph technologies that process data with contextual and conceptual intelligence. AllegroGraph is able to run queries of unprecedented complexity to support predictive analytics that help organizations make more informed, real-time decisions.

About Franz Inc.

Franz Inc. is an innovative technology company with expert knowledge in developing and deploying Graph Search solutions. AllegroGraph, Franz's flagship, high-performance, transactional, and scalable Graph Database, provides the solid storage layer for powerful Enterprise grade NoSQL solutions. AllegroGraph's Activity Recognition capabilities provides a powerful means to aggregate and analyze data about individual and organizational behaviors, preferences, relationships, plus spatial and temporal linkages between individuals and groups.

For additional Franz Inc customer success stories please visit:

- AllegroGraph – <http://franz.com/agraph/success/>
- Allegro CL – <http://franz.com/success/>

Franz's Professional Service team is in the business of helping companies turn Data into Information and Information into Knowledge. We combine Data, Business Intelligence, and Analytics consulting services under one roof for our customers.

Franz, an American owned company based in Oakland, California, is committed to market-driven product development, the highest levels of product quality and responsive customer support and service. Franz customers include Fortune 500 companies in the government, life sciences and telecommunications industries. Franz has demonstrated consistent growth and profitability since inception. For more information, visit franz.com.

All trademarks and registered trademarks in this document are the properties of their respective owners.