

December 2007



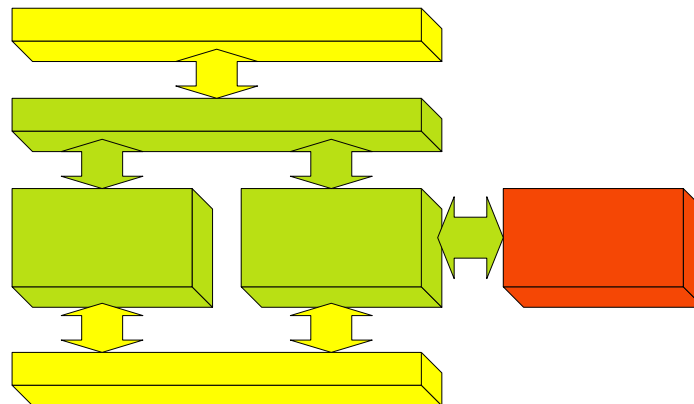
IBM Global Name Recognition Version 3.1

IBM NameWorks

*Frank Whaley
IBM Global Name Recognition*

Previous GNR products have been a collection of name analysis and comparison functions, available in multiple components with varying usage techniques. While these components offer tremendous flexibility, this component-based architecture has required developers to familiarize themselves with the peculiarities of each component and to understand how to integrate the components into the workflow of their applications.

IBM NameWorks addresses these difficulties by providing a readily accessible tool for name analysis and comparison, combining individual functions into a high-level interface. *IBM NameWorks* is an interface layer above individual GNR components, combining related functions into meta-functions and providing a unified and consistent programming interface.



***IBM NameWorks* provides a unified interface to GNR components.**

Rather than requiring multiple objects and data types for performing linguistic processing only two fundamental objects are needed, one included in each of the Analytics and Scoring component bundles:

- **Analytics**

The *Analytics* class includes all functions needed for evaluating a single name, including parsing, culture classification, genderization, variant generation and determining country-of-association information. These linguistic processes can be used individually or together; for example, the *analyze()* method performs all linguistic operations and produces a single combined result containing all analysis information for a name.

- **Scoring**

The *Scoring* class includes all functions needed to compare two names or to search for a name in one or more datalists, along with ancillary tasks such as date comparison and name categorization that may be used to refine search results. Preparation for searching (parsing and culture classification) can be performed separately or included in a search operation.

All analytics and scoring functions use the same form of data, and the results of one process can be used as input to a secondary process.

***IBM NameWorks* implements best practices for name analysis and comparison**

The most effective techniques for using GNR components are not always apparent, so *IBM NameWorks* uses what the GNR linguistic team considers the best approach. For example, genderization should be performed on individual

name phrases from left to right, stopping when the first valid genderization is found. Using the JNI interfaces for *NameParser* and *NameGenderizer* this process would look something like:

```
NameField gn = parser.parseName(name).getName(0).getGivenName();
Results results = new Results();
for ( int p = 0; p < gn.getNumNamephrases(); p++ )
    if ( genderizer.lookup(gn.getNamephrase(p).getTransformedText(),
                          results ) )
        break;
```

Using the Analytics module of *IBM NameWorks* this operation is performed with:

```
GenderData gd = analytics.genderize(name);
```

IBM NameWorks performs all preliminary and intermediate actions (transliteration to Romanized forms, parsing and culture classification) automatically, as needed for specific operations.

***IBM NameWorks* simplifies overriding behavior**

GNR components include several methods for controlling or extending their capabilities, such as custom parsing tokens or comparison parameters. These methods typically require programming, but *IBM NameWorks* allows specification of override data in a configuration file. Controlling or extending GNR functionality can be done by adding entries like:

```
Elouera=GivenName
```

to the configuration file, eliminating the need for custom programming.

***IBM NameWorks* makes searching easy**

A search operation run against multiple datalists typically requires a considerable amount of programming. *IBM NameWorks* provides a single method for transliterating, parsing, classifying, searching against multiple datalists and combining all matches into a single result:

```
SearchResults = scoring.search(transactionId, name,
                               datalists, strategy,
                               maxReply, minScore );
```

Information about available datalists and how to search them is also contained in the configuration file. Search engine locations and communication ports can be added, removed or modified without custom programming. *IBM NameWorks* will search multiple datalists in parallel to reduce overall search time.

***IBM NameWorks* is available on many platforms**

Both a Java interface and a web service interface are included. The Java interface can be used directly on any of the supported platforms and the web service interface can be used either locally or remotely in SOA environments. Any programming environment that can utilize web services can take advantage of the name analysis and comparison tools provided by *IBM NameWorks*. The Java interface can be used to build custom SOA applications.