NSRL DATA DICTIONARY

Gary E. Fisher

INTRODUCTION

This document describes the tables and data elements within domains used within the National Software Reference Library (NSRL) database. It provides guidance on how values in each data element are formulated and provides examples.

REFERENCES

Gallagher, Leonard, and Mark LaRoy Zimmerman, "NSRL Database Architecture, Version: Draft, Date: 05/25/00."

TABLES

1. *Application*: Contains information pertaining to each application or software package within the NSRL. Most of this information is generated by the user upon logging the package into the NSRL database. It includes elements, such as the application name, version, application type, etc. An example of an application is WordPerfect 6.0.

2. *Contact*: Contains information about the person or organization listed as the point of contact for questions about an application package. Example elements include the contact name, organization, telephone number, e-mail address, etc. An example contact entry might be "David Elliott / Microsoft / 800-555-1212 / dave.Elliott@microsoft.com."

3. *Event*: Records information about relevant events in the process of adding application information to the database. Defined by the user, examples of data elements include event code, event date/time, application ID of the application affected, etc. Events include logging-in an application or verifying the updated entries in the database, etc. An application may have multiple events associated with it over time.

4. *Event_Type*: For each event that can be applied to an application, a corresponding event type must be defined beforehand. A description of the event, the event name, an event ID, etc., are added to the database. Values in the Event_Type table may include the following: Received into inventory, Added to database, Hash codes generated, database entries verified. This is user defined.

5. *File_Execute*: For each application entered into the database, identifying information for each associated file within the application package is entered along with demographic information, such as the file size, file type, etc. Not user definable (generated by program.)

6. *Hashes*: Each file entered in the File_Execute table has four associated hash values. These hash values are stored in the Hashes table along with a pointer to the associated File_Execute table entry. Not user definable (generated by program.)

7. *Manufacturer*: Each application package has an associated manufacturer that produced the package. Before a manufacturer can be referenced by an Application table entry, an entry must be made in the Manufacturer table by the user. This entry includes elements, such as the full manufacturer's name, address, etc. The user enters this information. An example of a manufacturer entry may include "Microsoft / Microsoft Corporation / 1 Microsoft Way / Redmond / Washington / 99021 / USA .

8. *Media_Info*: This table identifies and records each item of electronic media on which the application program is delivered. Each media item consists of a set of files that can be run through the hash generator. The information is user defined and includes elements, such as media type, manufactured date, etc. An entry for a 3.5" diskette might include "3.5 / Installation diskette / 2000-07-20" along with a serial number that identifies the particular diskette.

9. *Operating_System*: The OPERATING_SYSTEM table represents the collection of distinct operating system platforms that may be able to execute an application. Sometimes a platform will be dependent upon an underlying machine architecture and sometimes it will not. When the machine architecture makes a difference, this table identifies each possibility as a distinct operating system platform. Each application is designed to execute on one or more application platforms. An operating system entry must exist before it can be referenced by an application entry. An operating system entry may include "Win3.1 / Windows 3.1 / Microsoft.

10. *Staff*: The set of NSRL staff members is recorded in the STAFF table, along with the role played by that staff member in the project and minimal contact information. Each staff member is identified by a unique StaffCode. It is important that each staff member understand the duties and responsibilities of the role they play in the project. Each staff member is responsible for ensuring that the events they are responsible for get properly recorded in the database. A staff entry must exist before it can be referenced in an event. A staff entry may include "Fisher / Gary / NIST / 100 Bureau Drive / Gaithersburg / Maryland / 20899 / x3275 / Project Leader."

11. *User-defined Classification*: Aperiodically, the need arises to add a new qualifier or access path to the database. This can be done without disrupting the current structure of the database by allowing the database administrator to add this new information in a user-defined classification scheme. For example, if a new code is needed to group applications by type of user, then a new classification can be added, such as "user," with predefined valid codes. The values of this new classification can be assigned by manual means or by an automated procedure that computes the value based on values of other elements within the database. The original structure of the database is not affected, but a new capability is added.

DATA ELEMENTS

Each data element is listed alphabetically and described in the following, along with examples of values. The table in which a data element can be found is listed in parentheses. Multiple tables are separated by commas.

1. AddressLine1 (Contact, Manufacturer): User-entered first line of contact or manufacturer mailing address. Example: 100 Bureau Drive.

2. AddressLine2 (Contact, Manufacturer): User-entered optional second line of contact or manufacturer mailing address. Example: Mail Stop 8970.

3. AppId (Application, Event, File_Execute, Media_Info): A unique numeric identifier assigned by the system automatically for each application package installed.

4. ApplicationType (Application): A 1-level taxonomy of application packages. Values include the following:
    a. Database – a data management package that may include database design, data entry, data edit, and reporting capabilities, such as dBase or Access
    b. Design – a package for developing architecture, layout, or formats
    c. Educational – teaching or self-help software
    d. Extensions – patches or additions to packages
    e. Financial – software for keeping track of expenses, accounts, sales, payments, checks, stocks, commissions, mortgages, and similar items, or for computing valuations based on combinations of these items
    f. Game – entertainment software
    g. Graphics – graphics or imaging software
    h. Multimedia – software that provides audio or visual output, animation of images, interactivity, etc.
    i. Null or no value – represents an unknown type that cannot be defined (Note that other types can be added to this list.)
    j. Operating System – a system control package, such as Windows 98 or Linux
    k. Presentation – a slideshow presentation package
    l. Scientific – applications that perform scientific computations, solve equations, perform matrix manipulations, compute statistics, perform forecasts, etc.
    m. Spreadsheet – a document preparation package utilizing a columnar layout, such as Excel or Quattro Pro
    n. Suite – a collection of different applications under a single umbrella framework or common set of access methods
    o. Utility – a productivity software package, such as file management, security, contact management, etc.
    p. Web Browser – a package for viewing information on the World Wide Web
    q. Word Processing – a document preparation package, such as WordPerfect or Word

5. AppName (Application): The full name of an application or package. For example, the full name of Windows NT is Windows NT 4.0.

6. BarCode (Application): If a bar code is attached to the packaging of the software, such as a part number or other identifying mark, the string represented by the bar code is entered as an identifying piece of information. Example: 1234AB567-890.

7. City (Manufacturer): The name of the city from the mailing address of the software manufacturer. Example: Seattle.

8. Comment (Application, Contact, Event, Media_Info, Operating_System, Staff): Text concerning any aspect of an entry that does not fit the requirements of any other data element may be entered. Often, this element is used to denote some point of interest concerning a relationship with another data element or value. Example: “See the entry for this application in the operating system table.”

9. ContactId (Contact, Event): A unique number identifier assigned to each contact entry in the Contact table. It is referenced by the Event table. It is generated by the data entry software as a serial number.

10. ContactType (Contact): A reference to the type of person or organization acting as a manufacturer, vendor, or other organizational contact. Required. Current values are—
    a. Court – to indicate that a library item is under subpoena to a court of law
    b. Law – to indicate that a library item is assigned to a law enforcement person or organization
    c. Vendor – to indicate that an application package was purchased from a vendor
11. Country (Contact, Manufacturer): The name of the country from the mailing address of the software manufacturer. Example: USA.

12. CRC32 (Hashes): The 32-bit Cyclic Redundancy Checksum computed for a specific file within an application. Generated by the NSRL hash routine.

13. CreateDate (File_Execute): The date-time string entered by the database software to indicate when the entry was posted to the database.

14. Description (Event_Type): A text string used to define the characteristics of an event, such as “An application has been logged into the NSRL database.” describes the event “Logged.”

15. Email (Contact, Staff): A text string that defines an address used for sending and receiving electronic mail on the Internet. Example: gary.fisher@nist.gov.

16. EventCode (Event, Event_Type): A short text string that refers to a specific event type and that will evolve over time. Values are –
    a. Received – An application package was received by the NSRL. Associated information will include the date received, the staff member who received it, and its condition when received.

b. Logged – Information from an application package was entered into the NSRL database, but the process is not complete.
c. Hashed – Hash codes have been generated for an application package and have been added to the NSRL database.
d. Verified – The information for an application package that has been entered into the NSRL database has been verified through manual or automated processes.

17. EventName (Event_Type): The name of an event as recorded in the database. Example: "Application logged-in to database."

18. EventOrder (Event_Type): An accounting/management feature to allow the ordering of events for presentation in reports and lists.

19. EventTime (Event): The date and time an event occurred as recorded in the database.

20. FileExt (File_Execute): A short text string that indicates a file type as part of the file name. Examples include COM, BAT, EXE, DAT, SCR, CFG, etc.

21. FileName (File_Execute): The complete name of a specific file. In systems that use path names (e.g., directory hierarchies) or file extensions (e.g., FileExt), these may be included. Examples include \Program Files\rebol\view\rebol.exe, winword.exe, /usr/bin/shx, etc.

22. FilePrename (File_Execute): The root portion of a file name. For example, in the file, \Program Files\rebol\view\rebol.exe, the FilePrename is "\Program Files\rebol\view\".

23. FileSize (File_Execute): The size of a file in 8-bit bytes as reported by an operating system utility or application documentation.

24. FileType (File_Execute): The type of data contained within a file defined by its format. The format may or may not be reflected in the file name. A WORD document file (with .DOC extension) contains a header record of information that is used by WORD to determine what type of document the file contains. Generally, this information is not human-readable, existing in a binary form. If the header indicates that the file contains a WORD document, then the rest of the file's contents are read by the WORD program using a specific format that is characteristic of WORD document files. Not all files can be typed in this manner, especially those that do not use file headers.

25. FirstName (Staff): The first name or given name of a staff member. Example: Gary.

26. Globalization (Media_Info): An indicator that specifies whether a set of files falls under the rules regarding international use. If a file is meant only for use in the U.S., then the globalization item contains the value "USA". A null value specifies that no

information is available. Other values are not currently defined. The list will undoubtedly grow over time.

27. HashId (File_Execute, Hashes): A unique serial number that links a specific file in the File_Execute table to the hash codes generated from that file in the Hashes table. Not user definable (generated by program.)

28. InputDate (All tables except those that can be derived from existing data): The datetime string added to each row in a table to indicate when the information was stored in the database. Used in creating an audit trail of transactions in the database. Not user definable (generated by program.)

29. Language (Application): The language in which the application is accessible. Values include common spoken/written language names, such as English, Spanish, etc. User defined during initial package information entry.

30. LastName (Staff): The surname of a staff member. Example: Fisher.

31. MD4 (Hashes): A 128-bit string generated by the NSRL Builder program from each file in an application package using the algorithm specified in IETF RFC 1320. Stored in the database as a 16-character string of hexadecimal digits. Not user definable (generated by program.)

32. MD5 (Hashes): A 128-bit string generated by the NSRL Builder program from each file in an application package using the algorithm specified in IETF RFC 1321. Stored in the database as a 16-character string of hexadecimal digits. Not user definable (generated by program.)

33. MediaId (Event, File_Execute, Media_Info): The MediaId domain is a collection of character string identifiers used to distinguish the disks, CD's, cards, or other media in an application package. The MediaId usually is paired with the AppId to uniquely identify a specific media item within the package. The identifier is arbitrary and is not considered to carry any semantic intent, but just for convenience the identifier might also hint as to the type of media, e.g. CD1, CD2, Floppy1, Floppy2, EXE1, EXE2, etc., or if the media is named, the MediaId might use the printed name on the media label to distinguish one media item from the other. The actual type of media is carried by the MediaType attribute in the MEDIA_INFO table.

34. MediaSerialCode (Media_Info): The MediaSerialCode domain is a set of string values meant to capture any identifying serial number or model information printed on a media label. Often a floppy or CD will have such information printed directly on the disk – separate from the paper label. For CD's this information in often stamped in a circular form near the center hole of the CD, either on the front side or the back side of the CD.

35. MediaType (Media_Info): The MediaType domain is a set of character string values used to identify the type of media. Initial values might be: Tape, Floppy-9", Floppy-5", Cassette, Floppy-3.5", CDROM, WEB, PC Card, etc. This domain is still under development to determine the best tags to use.

36. MfgCode (Application, Manufacturer, Operating_System): The MfgCode domain is a set of character string identifiers used to identify companies that manufactured or developed the software application under process, or the operating system platform under which the software is executed. This code is the primary key of the Manufacturer table. This code is not supposed to convey any semantic information, but for convenience it may be a short-cut of the company name, e.g., IBM, Oracle, Microsoft, Intuit, etc. The full legal name, and other identifiers for a company, are given in other attributes of the Manufacturer table.

37. MfgDate (Media_Info): The date of manufacture of the application package. If it cannot be determined, it may be left as a null value.

38. MfgName (Manufacturer): The full name of the organization that manufactured the application package. Example: Microsoft Corporation.

39. Multiplicity (File_Execute): A serial number assigned to individual files that exhibit the same file name. This occurs, for example, when the same library file name is stored in two or more different directories at the same time, or is used in more than one application. Multiplicity differentiates between the various files. The hashes for each of these files will be different.

40. Name (Contact, Operating_System): The full name of the contact person or organization when used in the Contact table, or the full name of the operating system when used in the Operating_System table. Example: Gary Fisher.

41. Oscode (Event, File_Execute, Operating_System): The OScode domain is a set of character string identifiers used to identify operating system platforms under which the application software might install differently. This code is the primary key of the Operating_System table. Other attributes of this table also identify the version and platform under which the operating system is executing, so the same OS software may receive different codes for each version and each platform on which it executes. This code is not supposed to convey any semantic information, but for convenience it may be a short-cut of the OS name and the platform on which it executes, e.g. HP-Unix6.5, MSNT4.0, etc. The official name, version, and platform are given by other attributes of the Operating_System table.

42. PackagedWithin (Application): In cases where multiple applications are packaged together under one umbrella package, each of the individual applications must be identified, as well as the umbrella package. In such cases, the individual applications are identified as belonging-to another package through the PackagedWithin item. The value of this item is the AppId to which the current application belongs. For example,

if WORD (AppId=6) belongs to Office Suite (AppId=2), then the PackagedWithin value for WORD would be 2 to indicate that WORD was packaged within Office Suite.

43. Platform (Operating_System): The Platform domain is a set of character strings that describe the platform or platforms under which the operating system executes with no differences. For example, Microsoft Windows 95 is supposed to operate exactly the same under all platforms built on one of the Intel Pentium processors. If that is the case, this attribute may say "All Pentium II processors". If at a later time, distributors of Microsoft operating systems have the right to extend or modify the software, then it may be necessary to distinguish each operating system by its platform manufacturer, e.g. Dell-MS2000, etc.

44. PostalCode (Contact, Manufacturer, Staff): This consists of a set of character strings that represent world-wide post office codes. In the U.S. this may be a 5- or 9-digit code. In other countries, it may be an alphanumeric code.

45. ProjectRole (Staff): The ProjectRole domain is a set of character string identifiers used to represent the various roles that a staff person may play under the NIST-NSRL project. Initially, these roles may be straight-forward, like Librarian, Analyst, Intern, etc. but they could evolve over time to become more specific.

46. ReportTime (Event): A datetime timestamp used to identify when a specific event was entered in the Event table. Not user definable (generated by program.)

47. SHA (Hashes): A 160-bit string generated by the NSRL Builder program from each file in an application package using the algorithm specified in FIPS 180-1. Stored in the database as a 20-character string of hexadecimal digits. Also known as SHA-1. Not user definable (generated by program.)

48. StaffCode (Event, Staff): A user definable code assigned to each staff entry in the Staff table. It is used as a shorthand method of referring to a specific staff member and may be made up of elements, such as the first name, last name, or employee ID number.

49. StateProv (Contact, Manufacturer): The StateProv domain is a set of character strings used to identify the political sub-unit within a country that is directly under the country government. In the U.S. this attribute will identify states, territories, or districts. In British commonwealth countries it will identify provinces. At least for the present, the NSRL database will use short English names for countries and their subunits; at some point in the future this may evolve to codes that are language independent.

50. SubUnitOf (Manufacturer): A particular manufacturing organization may be a sub-unit of another organization. When entered, the value denotes which organization is the parent of the entry. For example, SunSoft is a sub-unit of Sun Systems.

51. Telephone (Contact, Staff): The Telephone domain is the set of visible character strings used to represent world-wide telephone numbers. The NSRL project has not yet specified a format for telephone numbers, but for US numbers it should use the full 10-digit number with area code, with unit separations by a hyphen, e.g. 800-555-1212. Full international numbers may be written in the form: +ccc-ddd-dddddddd, where the plus sign is used to indicate that some local access numbers may need to be dialed first, followed by the country code, followed by the telephone number within that country. Hyphens, or spaces, are used for readability only, although in some cases a space may indicate that a pause is necessary before continuing with the number. For U.S. numbers, the international format would look like +1-800-555-1212. If a telephone number has an extension, then the extension can be indicated by a space followed by the x character, followed by the digits of the extension, e.g. 800-555-1212 x2345.

52. Version (Application, Operating_System): The Version domain is a set of visible character strings that identify the version of a software application or operating system. The version is meaningless without also having the name or other identifier for the product. The format of a version is vendor-specific. The database simply records whatever the vendor gives as a version number for the product.

53. WebURL (Contact): The URL domain is a set of visible character strings, each of which identifies a Uniform Resource Locator (URL) as specified by W3C. (See http://www.w3.org/Addressing/URL/5_BNF.html.) An example URL is http://www.nsrl.nist.gov/.