Handle.Net Version 8.1 Software Release Notes

HTTP JSON REST API

In the past, handle servers have allowed an HTTP interface which responded to requests in the native binary handle protocol of RFC 3652. Now requests can also be made using a REST API, with requests and responses expressed via JSON encoding.

Browser Admin Client

A new admin client runs in-browser and uses the new REST API to administer the handle server. This admin client runs as a modular extension as described below. New server setups will include this by default; existing servers will need to copy the admin.war file into a 'webapps' folder inside their handle server directory.

Modular Extension Framework

Handle servers now incorporate an embedded Java Servlet container. Java Servlet apps can be added to extend the functionality of the server. Such extensions can (optionally) expose a user interface over HTTP, and can register themselves as "transaction queue listeners" to act on each incoming transaction.

Handle Protocol

- **mintNewSuffix.** A new flag on create handle requests allows handles to be minted by the server. The handle in the create request is augmented with a random UUID before creation. The created handle is returned in the response.
- **overwriteWhenExists.** A new (since 7.1) flag on create handle and add handle value requests will cause the server to overwrite existing handle records or handle values, rather than failing. This allows "create or replace" semantics.
- **Protocol version negotiation.** Handle clients find the protocol version supported by a server in the server's HS_SITE value. That version may not reflect the newest version actually supported by the server. New clients will use that as a starting point but can negotiate with the server the actual supported version to be used in the server's response message or in an ongoing session.
- **Security improvements for authentication.** All security features are implemented using cryptographic primitives supported by the JRE.
    - **RSA.** Full support for servers using RSA rather than DSA keys (since 7.3.1).
    - **AES.** Default encryption is now AES-128.
    - **SHA-256, HMAC, PBKDF2.** Default signature hash is SHA-256; default message authentication code uses HMAC-SHA-256. Secret key authentication now uses PBKDF2-HMAC-SHA1 by default.

    - **Improved streaming responses.** Streaming messages used for replication (that is, the retrieve transactions and dump handles responses) now use TLS for increased security and also speed.
- **Mirroring dump resumption.** The "dump handles" operation used by new mirror servers to bootstrap their operation can now be interrupted (by transient network or server issues) and will resume automatically.
- **ListNAsRequest.** An administrative handle request type to request a list of NAs homed on a given handle server.

Global Handle Registry

- **DONA and MPAs.** The responsibility for the overall administration of the GHR has moved from CNRI to the DONA Foundation, a newly-established non-profit organization in Geneva, Switzerland (see http://www.dona.net). Multiple organizations, known as MPAs (Multi-Primary Administrators), have been authorized and credentialed by DONA to provide MPA GHR Services. CNRI will operate as an MPA. The GHR service definition 0.NA/0.NA is now distributed over multiple handle records reflecting the MPAs. A new handle record 0.0/0.0 represents the DONA trust root for signature verification.
- **HS_SERV.** Prefix handles can now have multiple HS_SERV values; clients will follow all of them, recursively, to construct the collection of sites which serve a given prefix.
- **Derived prefixes.** Some prefix handles can be accessed at handle services other than the GHR. If the prefix handle record for a prefix handle 0.NA/X.Y contains HS_SITE.PREFIX or HS_SERV.PREFIX values, those values indicate the handle server sites where clients can resolve prefix handles of the form 0.NA/X.Y.Z.

Handle Server

- **No index authentication.** Typically handle authentication requires an entity to be identified as an index:handle pair. We want to allow entities identified by only a handle without needing to specify an index. To this end existing tools and the existing protocol will accept authentication by entities with index 0; in the case of public key authentication, the server will figure out the actually used index and give the client the appropriate privileges. HS_ADMIN values can give administrative rights to entities with index 0, which allows those rights to an entity with any prefix; this also allows granting rights to entities using secret key authentication without an index.
- **Replication bugfix.** A longstanding bug in replication involving a race condition may have allowed failures of replication in rare cases. The problem would be manifest by the situation where a handle exists but is empty of any values in the mirror. This bug has been fixed.
- **0.SITE/status handle.** A handle server can respond to the resolution of the 0.SITE/status handle with some information about its system, including some basic facts about load, memory, and disk space. The information is now only available to administrative users ("status_handle_admins" in config.dct, or turn off with "enable_status_handle" = "no").

- **BdbjeTransactionQueue.** The transaction queue (used for replication) now uses Berkeley DB JE, removing a performance bottleneck noted with the older file-based transaction storage.
- **Atomic file access.** File based operations use an atomic file change mechanism which will prevent issues previously possible when a handle server is running on a disk which becomes full, or experiences other disk issues. (System administrators can add "file_write_no_sync" = "yes" to the server config to turn off syncing.)
- **Deprecation of backup/checkpoint.** Handle servers have supported a backup/checkpoint administrative request. We have not found this to be useful in practice. With the BDBJE handle storage (the default since v7) the handle storage can be backed up at the command line with a tool like rsync while the handle server is active. The backup/checkpoint requests and command-line tools will continue to work if they are part of your workflow, but we no longer encourage their use.
- **Logging of administrator identity.** The access.log format has been changed to allow the handle value identifying the administrator of administrative operations to be logged.

Handle Client / Java Client Library

- **Faster UDP timeouts.** The default UDP timeouts have been halved; clients will thus switch to TCP more quickly.
- **Session recovery.** Clients will notice and automatically recover from session timeouts and other issues more reliably, initiating a new session rather than returning an error.
- **HandleResolver.resolveHandle()** now returns an empty array for a no-values response, rather than an error.
- **HandleSignature API.** The API for generating signed handle values has been rewritten.

Tools

- **hdl-migrate-storage-to-bdbje.** Servers which have been around since v6 may still be using a legacy storage backend. This tool will migrate the storage to the Berkeley DB JE storage which has been the default since v7. It can also be used to migrate SQL-based storage to BDBJE.
- **hdl-keygen.** This tool will generate a new public and private key in the Handle protocol key file format.
- **hdl-keyutil.** This tool will take a private key and add or remove password encryption to the stored key file. It can also change the encryption to be compatible with the v6 or v7 key-encryption formats, if the same key file will be used with multiple software versions.
- **hdl-convert-key.** This tool will convert public and private keys in the Handle protocol key file format to or from a more standard PEM format.
- **hdl-admintool.** This tool (the Java Swing GUI admin client) has had a number of minor improvements, including: the addition of a "replace mode" where a handle being edited can be replaced in a single operation, rather than series of add/modify/delete values requests; and loading and saving handle records in a JSON format.

- **JAVA_HOME.** The handle tools scripts will now use the value of the JAVA_HOME variable to indicate which Java environment to use.
- **Signature service.** For purposes of signing handle records, it is possible, instead of storing the private keys on the same machine as the handle client, to store private keys on a secure machine on the local network via a network service which only uses the keys to sign handle records; all signatures are performed on that secure machine instead of the machine running the handle client.

Configuration

- **Java system property net.handle.configDir.** Normally clients configure themselves based on files in the .handle subdirectory of the user's home directory. If Java system property net.handle.configDir is set, it is taken to be the full path to an alternate directory to be used instead of .handle.
- **Client config.dct.** Since 7.2 client configuration can be done via a config.dct file in the .handle configuration directory. Supported configuration options include "tcp_timeout", "trace_resolution" = "yes", and "no_udp_resolution" = "yes", in addition to a few new configuration options listed below.
- **"ipv6_fast_fallback".** By default "ipv6_fast_fallback" = "yes" and clients will fallback to trying IPv6 and IPv4 in parallel if the IPv6 connection does not complete quickly. If "ipv6_fast_fallback" = "no" then clients will try IPv6 first and wait for an ordinary timeout before using IPv4 servers.
- **"max_idle_time".** Server configuration parameter for section "hdl_tcp_config" which specifies socket timeout; default five minutes.
- **"site_filter_keywords".** This client configuration option contains a string; when possible, the client will only talk to sites where there is a site attribute also called "site_filter_keywords" containing this string. Both the configuration option and the site attribute can be a whitespace separated list of tokens, and the resolver will prefer the site if its configuration option and the site's attribute contain a token in common.
- **"auto_update_root_info" = "no".** This option will prevent servers and clients from automatically updating their root info. The root info will need to be manually updated.
- **siteinfo.json.** Some configuration of a handle server is actually done via its stored copy of its siteinfo, which is also stored in the prefix handle as an HS_SITE value. As of version 7.2 this can be stored in a human-editable JSON file siteinfo.json, instead of the binary siteinfo.bin file. The tool hdl-convert-siteinfo can convert between the two formats.
- **bootstrap_handles.** Previously handle clients were configured with a binary .handle/root_info file containing information about the 0.NA/0.NA handle record which specified the GHR. This information is now in JSON format in a file .handle/bootstrap_handles, which may contain information about multiple handle records including 0.0/0.0, the DONA trust root, and the GHR Service handles for the various MPAs.
- **serverCertificate.pem.** A server will automatically generate a self-signed certificate to respond to HTTPS requests. The certificate is based on the usual server keypair of

pubkey.bin and privkey.bin. The certificate will be stored as serverCertificate.pem. Since contemporary browsers have deprecated DSA certificates, servers with DSA keys will use RSA instead, with an auto-generated private key stored as serverCertificatePrivateKey.bin. These files can also be manually populated with an HTTPS certificate and private key.

- **"auto_homed_prefixes".** This new server_config configuration value takes an array of strings, each of which is automatically homed on the server at startup if needed. This allows the prefixes for which a server is responsible to be partially controlled by configuration rather than requiring administrative requests to be performed by a client.
- **"replication_accept_prefixes".** A list of strings which can limit which handles a mirror is willing to replicate.
- **Windows 7.** Before v8 the handle software had trouble finding the user directory, and thus the .handle configuration directory, under Windows 7 and later. This bug has been fixed.
- **local_addresses.** This client configuration file allows a mapping of IP address to IP address; it is intended to deal with the case of handle clients and servers behind a NAT where the client is unable to talk to the server's public IP address, only a local IP address. As of v8 each line allows the two addresses to be whitespace separated rather than tab separated.
- **local_info.json.** This (advanced) client configuration file specifies a mapping from handle prefix to the site (or sites) which the client will use to request handles under that handle prefix. This is very rarely needed. In v7 and earlier this file had a special binary format; v8 allows a JSON format for easier editing.
- **Minor changes.** "bind_address", "backlog", and "max_handlers", which were previously required, can now be omitted and will be given sensible defaults.