HareScript API FAQs

Which libraries are expected to be stable? Can I invoke internal/undocumentated functions?

If an API is undocumented or in a library with 'internal' in its path there's a good chance that the feature is not considered to be stable or very useful outside the WebHare core. We recommend to reach out on the forum for advice and to assist defining and documenting a public version of the API (ideally using test).

But feel free to invoke the API - just be careful that there's a much bigger chance the API will change or go away in the future if noone knows you're using it.

An undocumented API may also be deprecated.

Future deprecations

We expect to deprecate or remove the following features in the future. You shouldn't use them on new projects anymore. If you cannot find an alternative, get in touch - we may reconsider deprecation or be able to transfer the code out of WebHare to a separate module.

Remember to also keep an eye on the changelogs for past and current actual deprecations

In general

- blexdev_forsmapi module is obsolete, use the Publisher Forms API
- socialite and google modules will be removed in the future
  - we're still looking for a good solution for maps
  - commonly used social APIs will probably be moved to system/webapi
  - configuration should be done inside modules themselves (eg registry keys) and not rely on socialite databases to store it
  - API key lookup for eg maps should be done using LookupAPIKey

Consilio

- The searchobject API (consilio/lib/search.whlib) will be removed. You should use RunConsilioSearch instead
- <meta name="consilio-" /> may be deprecated. Where possible, use pagelists to provide additional fields

- Spidering by Consilio may be deprecated. Where possible, use pagelists to provide a list of pages to index.

Publication and templates

- template-v2.whlib is obsolete and slowly being fully replaced by <webdesign> functionality
- harescriptfile-v2.whlib and content listings shouldn't be used at all anymore - use custom or prebuilt file types
- EmbeddedObjectBase is obsolete - use WidgetBase
  - And use <tabsextension> to implement the widget screen.
- <propertyeditor> in siteprofiles is obsolete - use <tabsextension>

System APIs

- MakeEmailComposer is obsolete - use PrepareMailWitty, see also Building emails

WRD APIs

- Everything using mod::wrd/lib/objectapi.whlib is obsolete.
  Use mod::wrd/lib/api.whlib for new code
  - As an intermediate step, switch from ->GetTypeByTAG("XX") to ->^xx. Whether or not you're dealing with an old or new style WRD schema, ^ will give you a 'new style' WRDType object.

DATA Faqs

How do I retrieve all files in a site?

You might try to do this by using a WHERE filter on *parentsite* or *whfspath,* but you'll find this to be too slow because these fields are calculated - and trying to use these fields in a WHERE clause might require the database to calculate the parentsite for **every** file in the database.

An efficient approach is to use the GetDescendantFolderids API on a WHFS object. This function will retrieve all folder ids from a specifed point (and requires only one database query per depth level of the site) and you can then use this with an **IN** query on the parent field. This field is indexed and queries on this field should execute much faster than on fields like parentsite

```
INTEGER ARRAY subfolders := OpenWHFSObject(siteid)->GetDescendantFolderids();
INTEGER ARRAY allfolders := [ siteid, ...subfolders ];

RECORD ARRAY allfiles :=
   SELECT id, name, title
     FROM system.fs_objects
    WHERE parent IN allfolders AND isfolder = FALSE;
```