Multisite improvements in WordPress 6.1

Using the metadata API for network options

The way that network options are accessed is changing in WordPress 6.1. Network options have always been stored in a `meta` table called `sitemeta`. This name is confusing, as “site” in this context means “network,” not to be confused with blog `meta`, used to store site metadata. Prior to WordPress 6.1, the functions `get_network_option`, `add_network_option`, `update_network_options`, and `delete_network_options` used custom database queries and caching. This resulted in some performance issues and lots of code to maintain. In WordPress 6.1, the network options functions have been updated to use the more established metadata API, which is already used for metadata of other object types like posts, terms, and users. This effectively makes a function like `get_network_option` a wrapper around `get_metadata`. This has a number of advantages including:

- Consistency with other metadata types
- Support for `register_meta` functionality, such as default values
- Improved cache priming
- Fewer database queries, as all network options are primed in a single request

Along with these improvements, `WP_Network_Query` has a new parameter called `update_network_meta_cache` that allows for all networks in the query to prime the network options in a single query.

One side effect of this change is that newly updated network options using an integer value may result in a string being returned. When using values like this with an integer comparison, it is important to allow cast to an int. In the old implementation, the second page load of the option would have resulted in returning a string. This change can be considered a fix but it is different from the current behavior.

This change makes cache group `site-options` no longer in use and a candidate to be removed in future releases.

For more info see Trac ticket: #37181

Store main site ID of network in network options

When a multisite is created in WordPress 6.1, the main site ID on the network is stored in network options. As the main site on the network does not change in most cases, this is a value that can be stored for future reference. This saves a call to `WP_Site_Query` to look up the main site by domain and path in the bootstrap process which makes This lookup wasteful and not needed. This change also helps setup multiple networks, as noted above, since network options are primed in a single request. This means that when looking up multiple networks, the main site id is now primed in along with other network options, resulting in fewer database queries / cache lookups.

For more info see Trac ticket: #55802