Extension load ordering

Introduction

In eZ Publish 4.4 and higher, there is a mechanism that enables extensions to define in which order they should be loaded in regards to other, related, extensions. The function automates the extension load ordering to help avoid situations where extensions does not work properly in case of wrong order.

Requirements

For the extension load ordering mechanism to be active, extensions should be recent enough to contain the needed metadata stored in the "`extension.xml`" file, so make sure you use the extensions certified for your eZ Publish version (see: Technical manual / <version> / Installation / Requirements).

Configuration

To activate the extension load ordering, the following configuration setting should be enabled in "`site.ini.append.php`" in setting/override:

```
[ExtensionSettings]
ExtensionOrdering=enabled
```

Extension ordering

The order in which active extensions appear in the "`site.ini`" under `<eZ_Publish_root_folder>/ezpublish_legacy/settings/override/site.ini` "`[ExtensionSettings] ActiveExtensions=`" is important for the selection of templates and, possibly, configuration settings. For this reason, users must insure that the correct order is used for extensions that have dependencies on other extensions. For example eZ Find must be activated before eZ Webin. The same rule applies for the eZ Teamroom extension: eZ Teamroom and all its dependent extensions, such as eZ Lightbox and eZ Event, need to be activated before eZ Webin.

Although extensions can be activated using the Administrator Interface, as for now the admin interface does not provide a way to re-order the extensions, manually or automatically. This implies that the order can only be defined in site.ini.

Defining the correct order can sometimes be tricky when many extensions need to be considered. By providing metadata to the extension loading, manual extension load ordering is now a thing of the past.

Metadata

By providing metadata for the extension load ordering we enable eZ Publish to correctly determine the correct order itself. The metadata about extension ordering is optional. These metadata are now declared using an XML file, that can also be used to replace what is currently in "`ezinfo.php`". This file is named "`extension.xml`", located at the root of the extension. It has the following content:

```xml
<?xml version="1.0" encoding="utf-8" ?>
 <software>
     <metadata>
         <name>Extension name</name>
         <description>Extension description</description>
         <version>X.Y.Z</version>
         <copyright>Copyright text</copyright>
         <license>License type</license>
         <info_url>http://url/to/extension/site</info_url>
         <author>Extension author</author>

         <software>
             <uses>
                 <name>Used library/software 1</name>
                 <license>Used library/software 1  license</license>
                 <copyright>Used library/software 1  copyright</copyright>
                 <info_url>Used library/software 1  URL</info_url>s
                 <version>Used library/software 1  version</version>
             </uses>
             <uses>
                 <name>Used library/software 2</name>
                 <license>Used library/software 2  license</license>
                 <copyright>Used library/software 2  copyright</copyright>
                 <info_url>Used library/software 2  URL</info_url>s
                 <version>Used library/software 2  version</version>
             </uses>

         </software>

     </metadata>

     <dependencies>
         <uses>
             <extension name="usedExtension" />
         </uses>
         <requires>
             <extension name="requiredExtension" />
         </requires>
         <extends>
             <extension name="extendedExtension" />
         </extends>
     </dependencies>

 </software>
```

The first block (software/metadata) replaces "`ezinfo.php`". The fields are equivalent to those found in the info() method of the 'old' ezinfo. The sub-node software/metadata/software replaces the previous 3rdparty_libraries entry. One software/metadata/software/uses node must be used for each library.

The second block (software/dependencies) contains 3 sub-nodes:

- **uses:** indicates that the referenced extension is used by this extension.

- **requires:** indicates that the referenced extension is required by this extension.
- **extends:** indicates that the referenced extension is overloaded by this extension.

Note: 'uses' and 'requires' currently have the same effect, but their behavior will change when real dependencies checking is developed in the future.

Know issue

1. Extension load ordering breaks possibility to define your cluster (`file.ini`) settings for siteaccess and extensions as the functionality loads cluster before extension and siteaccess settings are loaded.
2. Extension load ordering has a small impact on performance

In both cases work around is to disable extension load ordering.