

# Zend\_Controller\_Action\_Helper\_ContentNegotiation

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

## Zend Framework: Zend\_Controller\_Action\_Helper\_ContentNegotiation - Matthew Turland Component Proposal

<b>Proposed Component Name</b>	Zend_Controller_Action_Helper_ContentNegotiation - Matthew Turland
<b>Developer Notes</b>	<a href="http://framework.zend.com/wiki/display/ZFDEV/Zend_Controller_Action_Helper_ContentNegotiation+-%2D+Matthew+Turland">http://framework.zend.com/wiki/display/ZFDEV/Zend_Controller_Action_Helper_ContentNegotiation - Matthew Turland</a>
<b>Proposers</b>	Matthew Turland
<b>Zend Liaison</b>	TBD
<b>Revision</b>	1.0 - 17 September 2008: Initial Draft. (wiki revision: 3)

## Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

### 1. Overview

Zend\_Controller\_Action\_Helper\_ContentNegotiation is an action helper for facilitating content negotiation, which is useful for developing RESTful web services and web applications in general when requested resources may have multiple potential representations.

### 2. References

- [HTTP/1.1: Content Negotiation](#)
- [Content Negotiation - Apache HTTP Server](#)
- [RFC 2295 - Transparent Content Negotiation in HTTP](#)

### 3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** evaluate any Accept header included in the application request, accept manual additions of server-supported

- content types, and optionally integrates with `Zend_Controller_Action_Helper_ContextSwitch` to obtain those types if it is in use.
- This component **will** evaluate any `Accept-Language` included in the application request, accept manual additions of server-supported languages, and optionally integrates with `Zend_Locale` to determine the current language in use by the application.
  - This component **will** evaluate any `Accept-Charset` header included in the application request, accept manual additions of server-supported character sets, and optionally integrate with `Zend_View` to determine the current character encoding in use by the application.
  - This component **will** evaluate any `Accept-Encoding` header included in the application request, accept a manual specification of a server-supported content encoding, and use a `Zend_View` output filter to apply the selected encoding algorithm.
  - This component **will not** provide feature negotiation features described in RFC 2295 and 2296 in its initial iteration. These features **may** be added in the future depending on demand.

## 4. Dependencies on Other Framework Components

- `Zend_Controller_Action_Helper_Abstract`
- `Zend_Controller_Action_Helper_ContextSwitch` (optional)
- `Zend_Locale` (optional)
- `Zend_View`

## 5. Theory of Operation

User accesses the action helper normally through the helper broker within the action controller `preDispatch` hook method and explicitly provides any custom configuration settings. These include content types, languages, character sets, and content encoding. In the absence of custom configuration, optional integration with other components with other components is used to infer values. The action helper will then apply all necessary modifications to the response headers and view.

## 6. Milestones / Tasks

- Milestone 1: [Design notes will be published here.](#)
- Milestone 2: Working prototype checked into the incubator supporting use cases.
- Milestone 3: Unit tests exist, work, and are checked into SVN.
- Milestone 4: Initial documentation exists.

## 7. Class Index

- `Zend_Controller_Action_Helper_ContentNegotiation`

## 8. Use Cases

UC-01

... (see good use cases book)

## 9. Class Skeletons

```
class Zend_Controller_Action_Helper_ContentNegotiation extends
Zend_Controller_Action_Helper_Abstract {
}
```

```
]]</ac:plain-text-body></ac:macro>
]]</ac:plain-text-body></ac:macro>
```