The system provides a debugging tool that shows the synchronization activity of web farm servers. Web farm debugging allows you to troubleshoot web-farm-related issues and find out if the synchronization works correctly.

Enabling the web farm debug

To use web farm debugging, open the **Settings** application and adjust the settings in the **System -> Debug** category:

⚠ **Note**: Web farm debugging is only functional if at least one web farm server is defined.

Setting	Description
Enable web farm debug	Enables web farm debugging and the **Web farm** tab in the interface of the **Debug** application.
Debug web farm operations of UI pages	If enabled, the web farm debug includes operations performed via the administration interface. Web farm debugging must also be enabled.
Web farm debug log length	Sets the maximum length of the web farm debug log, i.e. the number of requests for which the debug stores information. If empty, the value of the **Default log length** setting is used.
Display stack information	If enabled, the system tracks the code stack when debugging web farm tasks and displays the information in the **Context** column. This information is only available in the debugging UI and on the live site, not in the debug log written into the *logwebfarm.log* file.
Log web farm operations to file	If enabled, the system saves the web farm debug log into the *logwebfarm.log* file in the *~/App_Data* folder. This option does not require web farm debugging to be enabled.

✓ **Tip**: You can also enable web farm debugging through the "debug everything" settings in the **All** section of the **Debug** settings category.

Viewing the web farm debug

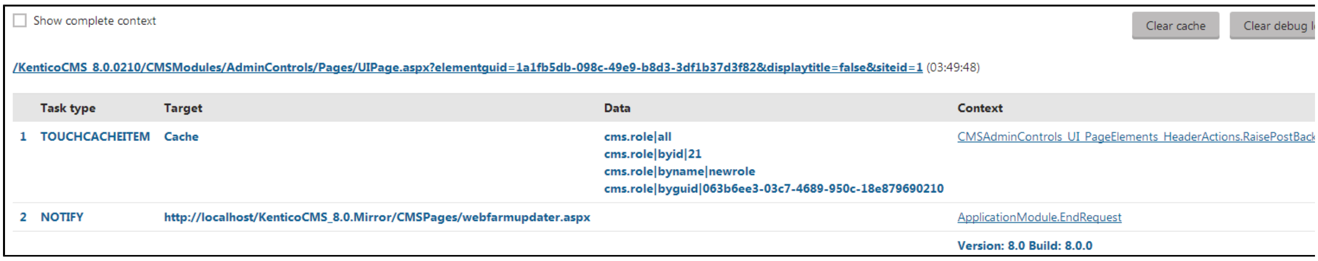
To access the web farm debug, open the **Debug** application and select the **Web farm** tab.

The log displays the following information for the current web farm server:

- Synchronization tasks created by the server for recent web requests
- Notifications sent to other web farm servers (**NOTIFY** Task type)
- Tasks processed as a result of notification requests from other servers in the web farm (**DO:** prefix in the Task type)

For example, when a web farm server handles a request that adds a new role object, the debug log shows:

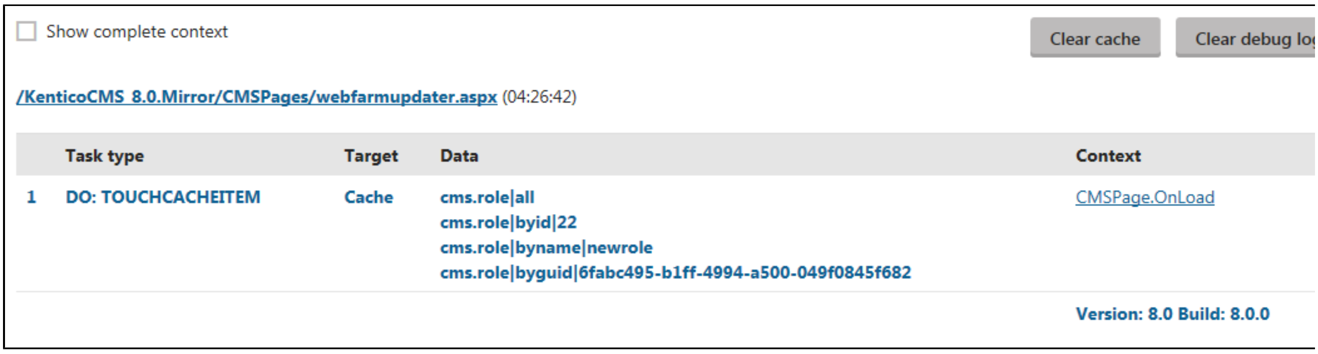
- The creation of a **TOUCHCACHEITEM** synchronization task (clears cached role objects)
- **NOTIFY** tasks for other servers in the web farm



/KenticoCMS_8.0.0210/CMSModules/AdminControls/Pages/UIPage.aspx?elementguid=1a1fb5db-098c-49e9-b8d3-3df1b37d3f82&displaytitle=false&siteid=1 (03:49:48)

	Task type	Target	Data	Context
1	TOUCHCACHEITEM	Cache	cms.role|all cms.role|byid|21 cms.role|byname|newrole cms.role|byguid|063b6ee3-03c7-4689-950c-18e879690210	CMSAdminControls_UI_PageElements_HeaderActions.RaisePostBack
2	NOTIFY	http://localhost/KenticoCMS_8.0.Mirror/CMSPages/webfarmupdater.aspx		ApplicationModule.EndRequest
				Version: 8.0 Build: 8.0.0

On the target web farm servers, the debug shows the processing of the synchronization task:



/KenticoCMS_8.0.Mirror/CMSPages/webfarmupdater.aspx (04:26:42)

	Task type	Target	Data	Context
1	DO: TOUCHCACHEITEM	Cache	cms.role|all cms.role|byid|22 cms.role|byname|newrole cms.role|byguid|6fabc495-b1ff-4994-a500-049f0845f682	CMSPage.OnLoad
				Version: 8.0 Build: 8.0.0

To remove all records previously logged in the web farm debug, click **Clear debug log**.