

KWizCom Corporation



SharePoint Enterprise Aggregation Caching Feature

User Guide

Copyright © 2005-2021 KWizCom Corporation. All rights reserved.

Company Headquarters

95 Mural Street, Suite 600
Richmond Hill, ON
L4B 3G2 Canada

E-mail: info@KWizCom.com

Web site: <http://www.KWizCom.com>

Sales

E-mail: sales@KWizCom.com

Telephone: +1-905-370-0333

CONTENTS

Contents	3
introduction	4
General	4
Product Overview	5
Solution Components	7
Cache implementation	7
Key Features	9
Installation Guide	10
Software Requirements	10
Server Requirements	10
Client Requirements	10
Installation Procedure	10
post Installation	10
Un-Installation Procedure.....	11
Administration Guide	12
Feature Initial setup.....	13
Creating and managing aggregation caching rules.....	16
Creating a new aggregation caching rule	17
Managing existing aggregation caching rules.....	21
Manually running an aggregation caching rule	23
Aggregation caching job log.....	24

INTRODUCTION

GENERAL

This document provides all the information needed to install, evaluate and deploy this KWizCom product:

- Introduction
- Installation Guide
- Administrator Guide

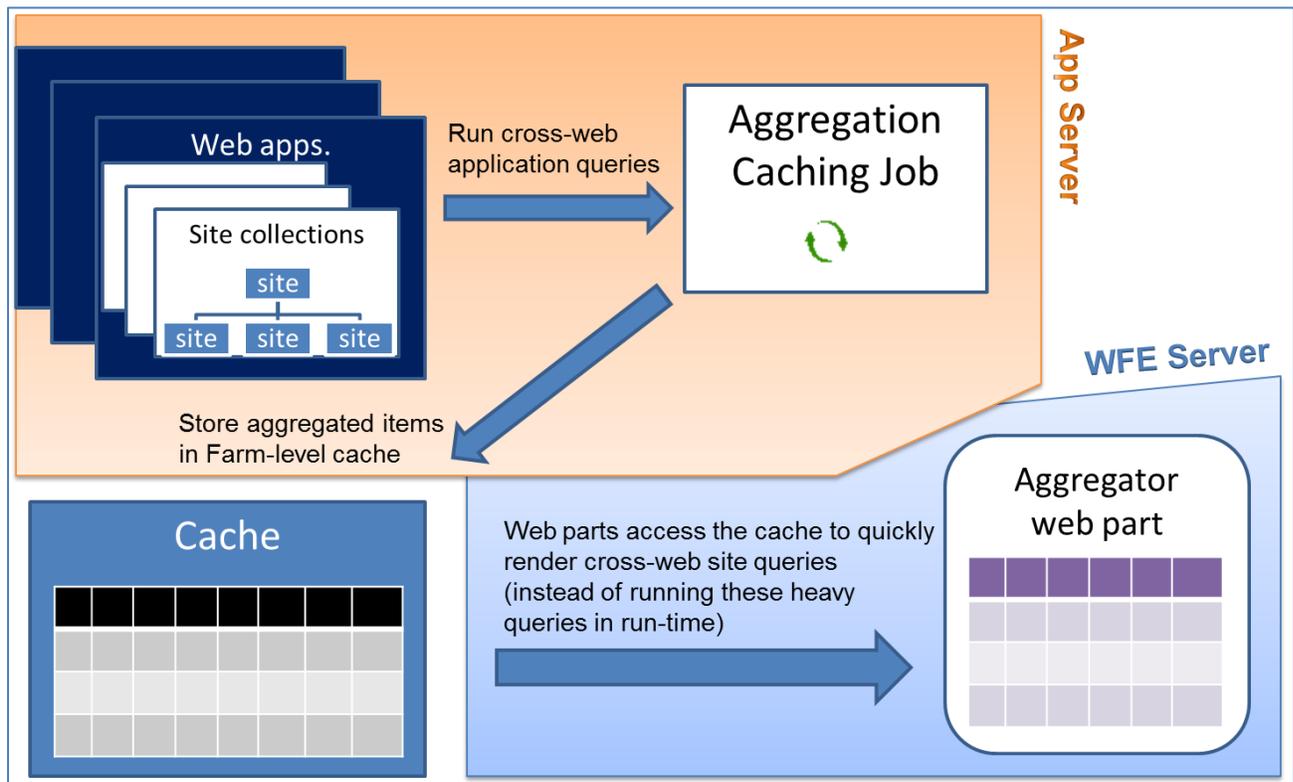
PRODUCT OVERVIEW

Boost Multiple Web Application Aggregation Performance!

Many customers require cross site collection/cross web application aggregation to retrieve business information stored in SharePoint. When trying to implement such aggregation capabilities, there is a need to overcome the following challenges:

- **No API for cross site collection queries**
SharePoint provides an API only for cross site queries. This means that cross S.C queries require custom implementation (development).
- **Cross site collection queries' performance degradation**
Aggregation of several S.C's involves separate queries (each for every S.C), and then a merging operation of the results. This means that the performance is linearly-dependent on the number of site collections, and also of the merged data.
- **Memory usage in cross site query**
Even running a simple SPSiteDataQuery to get information from a single site collection can be very expensive in server resources such as CPU, and memory. Depending on the number of lists and sites that are participating in the query, this single line of code can easily allocate 400MB, 600MB or more from the available RAM of the WFE server. Having 2 users running the same query at the same time will double that memory allocation just for them. This can sometimes result in crashing the entire application pool, affecting the overall performance of the server, and the only solution for this (query throttling to limit the number of items in queries) may resolve this issue but affect the results.
- **Heavy aggregations affect other user's browsing experience**
Running aggregation that take a long time and consume server resources affect all other users. This is because it is executed by the same application pool that serves all other users browsing the SharePoint WFE.

The KWizCom's *Aggregation Caching Feature* solves these issues by providing a centrally-managed aggregation mechanism. That mechanism allows administrators to configure the requested aggregations, and to store the results in a farm-level cache.



This allows the KWizCom List Aggregator web part (professional edition) and other consumers (using public API) to use the cached data to display aggregation results to users without gathering them in real-time.

The caching feature instead runs on a monitored SharePoint Timer Job which can be run on a dedicated application server, leaving your WFE free to service end user requests, as intended.

The timer job produces a per-rule report and a “last job run” report, which allows administrators to analyze and optimize the aggregation rules, as well as to monitor issues and identify rules that are taking too long to run. Administrators can easily disable a specific rule from being updated (temporary or permanently) and subscribe to alerts when rules report errors during update, or when rules take longer than a certain amount of minutes to run.

SOLUTION COMPONENTS

The solution includes the following modules:

1. **Aggregation Caching Job Feature**

This Farm feature creates the job and supporting backend list that stores all the aggregation rules.

2. **Aggregation Caching Rules Settings list**

This central list is used to store and manage the aggregation caching rules. Each aggregation caching rule defines a query (aggregation). An aggregation caching rule includes:

- a. A query
- b. Scope
- c. User account used to run the query
- d. Period of running the query (and refreshing the cache)

3. **Aggregation Caching Job**

This job runs the aggregation caching rules, and stores the results in the cache. Each rule can be executed by a thread, or serially one after another by the job.

CACHE IMPLEMENTATION

Once an aggregation caching rule has been executed by the job, its returned results are cached in the following way:

1. The results are merged (in case a rule includes several sources that need to be aggregated) into a DataTable object (in memory), along with other supporting information (such as, column ordering, grouping and more).
2. Serialize the object into XML, and compress it using ZIP.
3. Save the compressed file as an attachment to the rule list item.

When the List Aggregator web part (or any other consumer) needs to display cached results of selected Aggregation Caching Rule, it performs the following:

1. De-compress the compressed file attached to the caching rule's list item.

2. De-Serialize the XML file and load data to a DataTable object.
3. Bind the web part's grid display to the DataTable object.

These steps occur on every end user operation, which requires the List Aggregator web part to refresh the display:

- Sorting
- Filtering
- Page navigation
- Grouping
- Refreshing the entire page

This means that whenever the cache is refreshed (by the aggregation caching job, or manually by a user), any of those user-actions will cause the web part to display the results from the latest cache.

KEY FEATURES

Feature	Enterprise Aggregation Caching Feature	SharePoint 2013-2019
Enable configuring cross site collection and cross web application queries	√	
Execute aggregations by a central job that can be deployed apart from WFE servers that serve the users	√	
Cache aggregation results in a farm-level cache	√	
Cache is accessible through API for custom web part to use	√	
Enable end-users to manually refresh the Aggregation cache	√	
Centrally manage and monitor aggregation rules to better protect your server resources	√	
Works with KWizCom List Aggregator web part (Professional Edition).	√	

INSTALLATION GUIDE

SOFTWARE REQUIREMENTS

SERVER REQUIREMENTS

- SharePoint 2013-2019 server

CLIENT REQUIREMENTS

- Windows 10+
- Microsoft Internet Explorer/Edge, Mozilla Firefox, Google Chrome - Latest versions

INSTALLATION PROCEDURE

1. Log in as SharePoint administrator account to your SharePoint front-end server.
2. Browse the product's page on KWizCom's website and click the "Go" link under the "Install using our Web Installer" section.

This will start the KWizCom web installer.

Note: The KWizCom Web Installer will automatically offer you to upgrade other existing KWizCom products on your farm, in case you have old versions installed. You can check which products you wish to update at this time.

3. In case you prefer just to download the relevant .wsp packages and manually deploy them, then in the web installer UI check the "Download only" option (3rd page in the installer's wizard). When the installer finishes you will be able to go to the installer's folder on your server, and install the KWizCom packages by simply running the "SP-WSPInstaller" PowerShell script.
4. At the end of the installation process, you should see all packages deployed in the **Central Admin >System settings>Manage Farm Solutions** page, and the KWizCom SharePoint Foundation package should be deployed also in the Central Admin's web application.

POST INSTALLATION

After completing the package installation and deployment, activate the *SharePoint Aggregation Caching Farm* feature in Central Admin.



KWizCom Enterprise Aggregation Caching Feature

Activate this feature to enable caching service to enhance performance of cross site collection and cross web application aggregations

Deactivate

Active

UN-INSTALLATION PROCEDURE

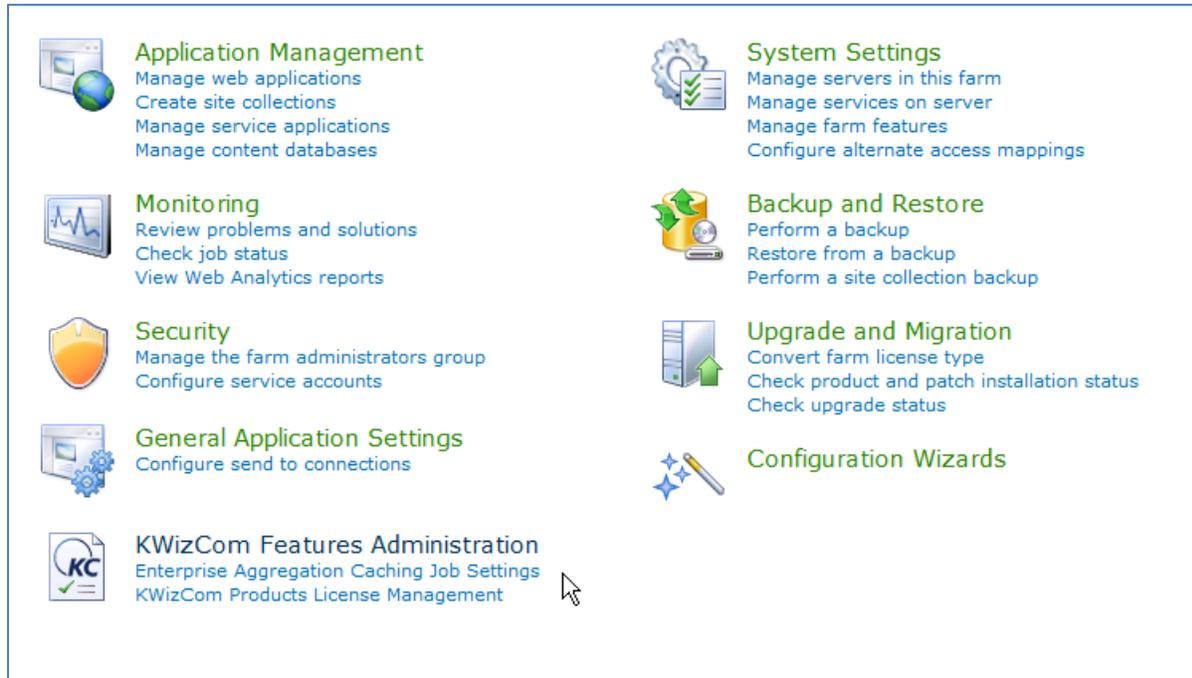
Retract the solution using SharePoint central admin or by using stsadm/powershell command line.

ADMINISTRATION GUIDE

This section describes all administration activities related to KWizCom Aggregation Caching feature, including: Product license activation, initial setup, configuration, and maintenance.

FEATURE INITIAL SETUP

After you install & deploy the product, and activate the farm feature (see “Post Installation” section), you will see a new link in the Central Admin main page, called: “Enterprise Aggregation Caching Job Settings”:



The screenshot displays a grid of management categories in the Central Admin interface. Each category includes an icon and a list of sub-options:

- Application Management** (Icon: Computer monitor with globe)
 - Manage web applications
 - Create site collections
 - Manage service applications
 - Manage content databases
- Monitoring** (Icon: Line graph)
 - Review problems and solutions
 - Check job status
 - View Web Analytics reports
- Security** (Icon: Shield)
 - Manage the farm administrators group
 - Configure service accounts
- General Application Settings** (Icon: Computer monitor with gears)
 - Configure send to connections
- KWizCom Features Administration** (Icon: Document with 'KC' and checkmark)
 - Enterprise Aggregation Caching Job Settings
 - KWizCom Products License Management
- System Settings** (Icon: Gear with checkmark)
 - Manage servers in this farm
 - Manage services on server
 - Manage farm features
 - Configure alternate access mappings
- Backup and Restore** (Icon: Backup tape)
 - Perform a backup
 - Restore from a backup
 - Perform a site collection backup
- Upgrade and Migration** (Icon: Server rack with green arrow)
 - Convert farm license type
 - Check product and patch installation status
 - Check upgrade status
- Configuration Wizards** (Icon: Magic wand with stars)

Click this link, you'll be redirected to the following page:

Central Administration > KWizCom Aggregation Caching Job Settings

Use this page to create the Aggregation Caching List that will hold all the aggregation caching rules, processed by the caching job.

Web Application Select a web application.	<input type="text" value="http://kwizcom/"/>
Site Collection Select a site collection.	<input type="text" value="http://kwizcom"/>
Web Site Select a Web Site.	<input type="text" value="Sub1"/>
Enable multi-threading By default, caching job runs each rule in separate thread to maximize resource usage of your server. In some cases, depends on server load, this may return unexpected errors in caching results. In that case, you can turn off multi-thread support. Possible reasons are usually limited server resources such as CPU and memory.	<input checked="" type="checkbox"/> Enable multi-thread support

This page is used to configure the following initial settings:

- Create the Aggregation Caching Rules list – this is done by configuring the location where you wish this list to be created. Simply select the web application, site collection and web site where the list will be created.
- Enable multi-threading – the Aggregation caching job can run the aggregation caching rules in 2 ways:
 1. Multi-threaded – Aggregation caching rules are executed in parallel by allocating each its own thread.
 2. Single-threaded – Aggregation caching rules are executed one after another.

If server resources allow it, you should check the “Enable multi-thread support”, which provides better response. (but consumes more server resources)

After saving the configured settings in the 1st time, the following will happen:

1. The Aggregation caching rules list will be created in the defined SharePoint site.
2. The *KWizCom Enterprise Aggregation Caching* job will start every 1 minute and check if there are any active aggregation caching rules in the Aggregation Caching Rules list that it needs to execute.

That's it, initial setup is complete!

You can now proceed to creating new aggregation caching rules.

CREATING AND MANAGING AGGREGATION CACHING RULES

Aggregation caching rules are simply list items in the Aggregation caching rules list, where each defines an aggregation that is executed by the Aggregation caching job.

After the job executes an aggregation caching rule it serializes the result-set into an xml file. The xml file is then compressed and saved as an attachment in the Aggregation rule list item (and this is the cache).

The following sections describe the administration tasks of creating and managing these aggregation caching rules.

CREATING A NEW AGGREGATION CACHING RULE

Add a new item to the *Aggregation caching rules* list and the following New Item form will show up:

Title *	<input type="text"/>
List type *	<input checked="" type="radio"/> <input type="text"/> <input type="button" value="v"/> <input type="radio"/> Specify your own value: <input type="text"/> Please choose the list type you wish to aggregate. For custom list types please type in the list definition as a number. You may enter a title for convenient in this format: 40100 WikiPlus Library.
Site naming pattern	<input type="text"/> If specified, Aggregation will only load results from sites containing the text here. You can use * wildcard. Separate more than one using ;
List naming pattern	<input type="text"/> If specified, Aggregation will only load results from lists containing the text here. You can use * wildcard. Separate more than one using ;
Aggregation view url *	<input type="text"/> Please enter a list view url. The caching job will use this view's sort, filter, group, and viewed fields definitions to build cached data.
Aggregated sites *	<input type="text"/> Fill out the addresses of the aggregated sites, followed by the aggregation scope. The input should be in the following pattern: Url1,Scope1 Url2,Scope2 ... Where: Url = url of a site Scope = 0, 1, 2 or 3, where: 0 - selected site 1 - Selected site and sub-sites 2 - Entire site collection 3 - Entire web application
Aggregation user account *	<input type="text"/>   The aggregation will use this user's permissions to retrieve items.
Aggregation period *	<input type="text" value="20"/> <input type="button" value="v"/> How many minutes should pass before rebuilding the cache?
Allow manual cache refresh	<input checked="" type="checkbox"/> Check this if you want to allow end users to request the cache to be updated. If the property is unchecked, this option will not be available in the web part.
Start time	<input type="text"/> Enter the time to start running this job (hh:mm) in 24Hrs format, or leave it empty to run it from 00:00.
End time	<input type="text"/> Enter the time to stop running this job (hh:mm) in 24Hrs format, or leave it empty to run it untill 00:00.
Active	<input type="checkbox"/>

You need to fill-out the following fields:

Field	Description
Title	Provide a descriptive name for the rule (a name that describes what this rule aggregates). Example: "Milestones awaiting approval"
List Type	Select the type of list that you wish to aggregate. The drop-down control displays all existing out-of-the-box list definitions. You can also type a custom list definition (one that you've developed), in the following format: [List Definition ID] [Title]. Example: if you wish to aggregate custom Wiki Plus libraries (part of KWizCom's Wiki Plus solution), you should type: "40100 WikiPlus"
Site naming pattern	In order to further restrict you aggregation scope, you can define a site naming pattern. This will instruct the aggregation job to look only for sites that comply with that naming pattern and only items in these sites will be aggregated. You can include "*" in your pattern. Example: type "Project*" to have this rule aggregate only sites that their name starts with "Project" string.
List naming pattern	In order to further restrict you aggregation scope, you can define a list naming pattern. This will instruct the aggregation job to look only for lists/libraries that comply with that naming pattern and only items in these lists will be aggregated. You can include "*" and "?" wildcard in your pattern. Example: If you wish to aggregate tasks, but you need to aggregate items only from specific tasks lists, you can simply type these lists' names, separated by ";"
Aggregation view url	Type the url of a list view that will be used to query (aggregate) items. You can create that view wherever you want. This view is used as a query definition, according to its defined columns, sorting, filtering, grouping and item limit configurations.
Aggregated sites	This field defines the scope of aggregation. Fill out the addresses of the aggregated sites, followed by the aggregation scope. The input should be in the following pattern: Url1,Scope1 Url2,Scope2 ... Where:

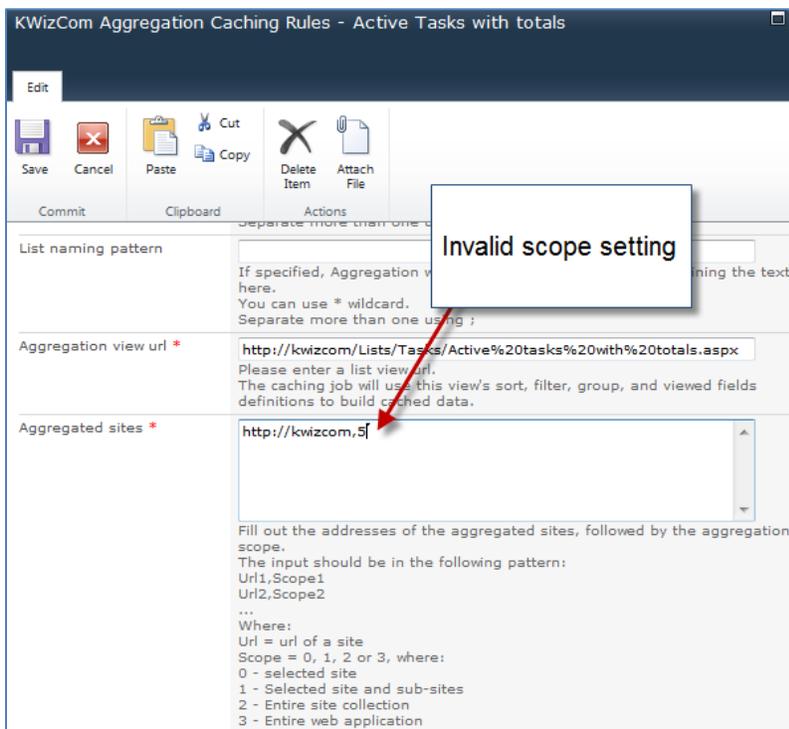
	<p>Url = url of a site</p> <p>Scope = 0, 1, 2 or 3, where:</p> <ul style="list-style-type: none"> 0 - selected site 1 - Selected site and sub-sites 2 - Entire site collection 3 - Entire web application <p>Example:</p> <p>If you wish to aggregate items across your entire farm, you should include all web applications:</p> <p>http://wepapp1, 3</p> <p>http://wepapp2, 3</p> <p>..</p> <p>http://wepapp3N, 3</p>
Aggregation user account	The aggregation will use this user's credentials to retrieve (aggregate) items. Use an account that has at least Read permissions, otherwise you will get "Access Denied" errors and the aggregation will fail.
Aggregation period	Select the period (minutes) of this rule execution. A rule should be executed no less than every 5 minutes.
Allow manual cache refresh	<p>Check this property if you wish to allow end-users to be able to manually refresh the cache.</p> <p>If this property is checked, end-users viewing aggregated data by using the KWizCom List Aggregator web part (Professional Edition), will be able to click the "Refresh Cache" icon to trigger a manual cache refresh.</p>
Start time	<p>You can define a time frame in which this rule will be processed by the Aggregation Caching job.</p> <p>Enter the time to start running this job (hh:mm) in 24Hrs format, or leave it empty to run it from 00:00.</p>
End time	<p>You can define a time frame in which this rule will be processed by the Aggregation Caching job.</p> <p>Enter the time to stop running this job (hh:mm) in 24Hrs format, or leave it empty to run it until 00:00.</p>
Active	Check this property to make this Aggregation Caching rule active.

The aggregation caching job processes active rules, and executes the ones that should be executed according to their defined period and last run time.

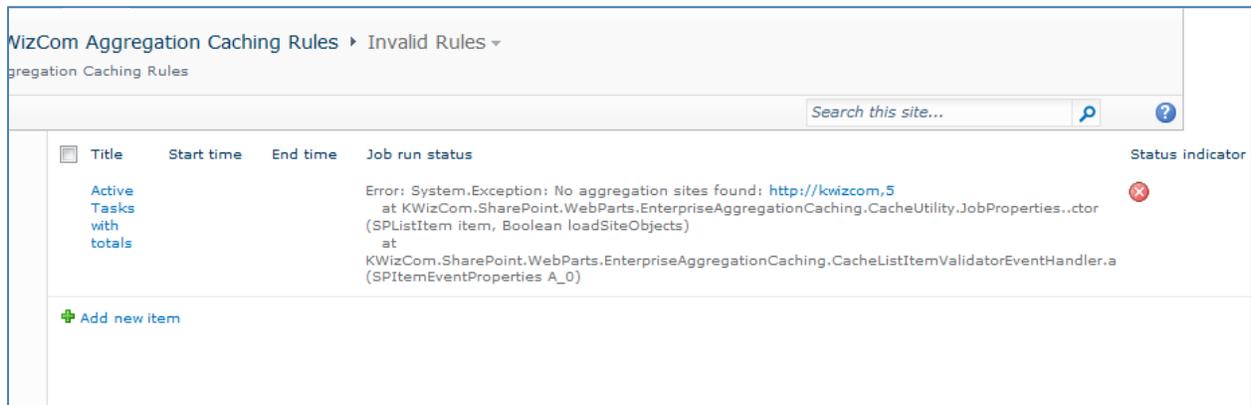
After you fill out and save the new aggregation caching rule item, it is validated by an event handler. If there are errors found, the new aggregation caching rule item will be in error status, and you will see it under the “Invalid Rules” list view.

Example:

The following screenshot shows an aggregation caching rule with invalid settings for the “Aggregation sites” field: (it has a scope value of 5 which is invalid)



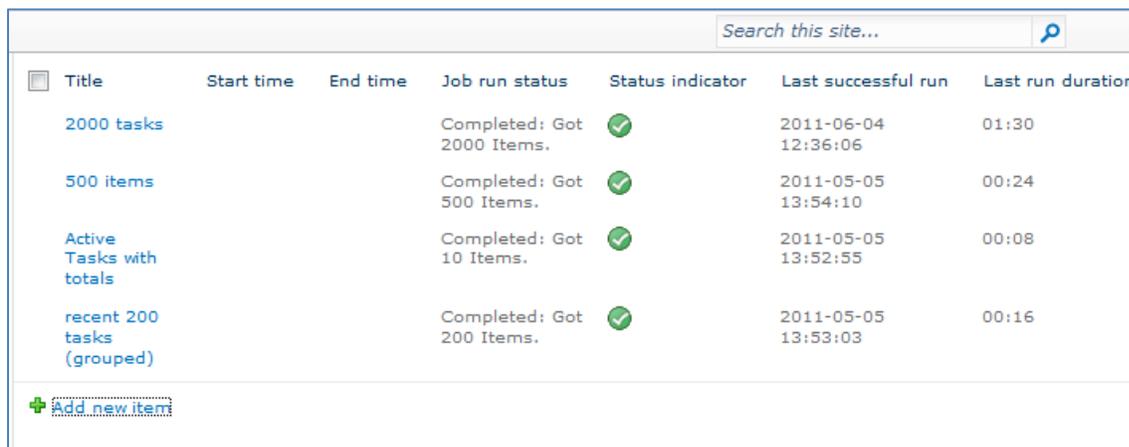
When saving this Aggregation caching rule, it will appear in the “Invalid Rules” view and will become inactive:



The “Job run status” field will show you the exact error details (both for validation errors and for aggregation run-time error).

MANAGING EXISTING AGGREGATION CACHING RULES

The management of aggregation caching rules is all done through the aggregation caching rules list.



Title	Start time	End time	Job run status	Status indicator	Last successful run	Last run duration
2000 tasks			Completed: Got 2000 Items.	✓	2011-06-04 12:36:06	01:30
500 items			Completed: Got 500 Items.	✓	2011-05-05 13:54:10	00:24
Active Tasks with totals			Completed: Got 10 Items.	✓	2011-05-05 13:52:55	00:08
recent 200 tasks (grouped)			Completed: Got 200 Items.	✓	2011-05-05 13:53:03	00:16

Looking at the “Active Rules” view, you can see the rules that are being periodically executed by the Aggregation caching job.

For every aggregation caching rule in this list you can see its status and last run details such as:

- How many items were aggregated
- When was the last successful run of the rule
- How long it took to execute the rule.

By monitoring this list you can see which aggregation caching rules consume more resources (longer execution time), and change their execution time, scope to match your requirements.

MANUALLY RUNNING AN AGGREGATION CACHING RULE

As mentioned before, aggregation caching rules are periodically executed (each according to its configured run period) by the Aggregation Caching job.

As an administrator, you can also manually run a required aggregation caching rule (if you wish to manually refresh the cache or you wish to test that aggregation caching rule).

To manually run an aggregation caching rule, simply edit the rule. In its edit form you will see at the bottom of the form the “Queued for manual update” checkbox:

The screenshot shows the edit form for an aggregation caching rule. The form includes the following fields and controls:

- Allow manual cache refresh:** A checkbox that is checked. Below it, text reads: "Check this if you want to allow end users to request the cache to be updated. If the property is unchecked, this option will not be available in the web part."
- Start time:** An empty text input field. Below it, text reads: "Enter the time to start running this job (hh:mm) in 24Hrs format, or leave it empty to run it from 00:00."
- End time:** An empty text input field. Below it, text reads: "Enter the time to stop running this job (hh:mm) in 24Hrs format, or leave it empty to run it until 00:00."
- Active:** A checked checkbox.
- Queued for manual update:** An unchecked checkbox, which is highlighted with a red arrow.
- Attachments:** A list containing "CacheXml.compressed" with a "Delete" button next to it.
- Version:** 2087.0
- Created at:** 3/29/2011 3:40 PM by KWIZ\administrator
- Last modified at:** 4/6/2011 2:24 PM by System Account
- Buttons:** "Save" and "Cancel" buttons.

Check that property and click “Save”.

The rule should be executed by the Aggregation Caching job as soon as it wakes up (The aggregation caching job starts every 1 minute).

After few minutes refresh the Aggregation caching rules list view to see the updated status of the executed rule.

AGGREGATION CACHING JOB LOG

The aggregation caching job starts every 1 minute and processes all active aggregation caching rules.

During each such run cycle, the job logs its executions steps and issues that occur during that cycle. You can the last job's run log in the *Enterprise Aggregation Caching Job Settings* page (in Central Admin):

<p>Enable multi-threading</p> <p>By default, caching job runs each rule in separate thread to maximize resource usage of your server. In some cases, depends on server load, this may return unexpected errors in caching results. In that case, you can turn off multi-thread support. Possible reasons are usually limited server resources such as CPU and memory.</p>	<input checked="" type="checkbox"/> Enable multi-thread support
<p>Last operation log</p> <p>Log from last job run</p>	<p>KWizCom Enterprise Aggregation Caching</p> <hr/> <p>5:36:35 PM: Job started 5:36:35 PM: Got jobs list 5:36:35 PM: Got active rules view 5:36:35 PM: Got 3 active rules from view 5:36:35 PM: Queue thread for rule: 3 5:36:35 PM: Skipping item 2000 tasks (3). Rule is scheduled for next run at 5:46:06 PM 5:36:35 PM: Queue thread for rule: 4 5:36:35 PM: Skipping item 500 items (4). Rule is scheduled for next run at 2:04:10 PM 5:36:35 PM: Queue thread for rule: 2 5:36:35 PM: Waiting for 1 threads to finish... 5:36:35 PM: Skipping item recent 200 tasks (grouped) (2). Rule is scheduled for next run at 2:03:03 PM 5:36:45 PM: Still waiting for 0 threads to finish... 5:36:45 PM: All threads finished running.</p> <hr/> <p>5:36:45 PM: Job finished</p>
<p style="text-align: right;"> <input type="button" value="OK"/> <input type="button" value="Cancel"/> </p>	