KWizCom Enterprise Aggregation Caching Feature

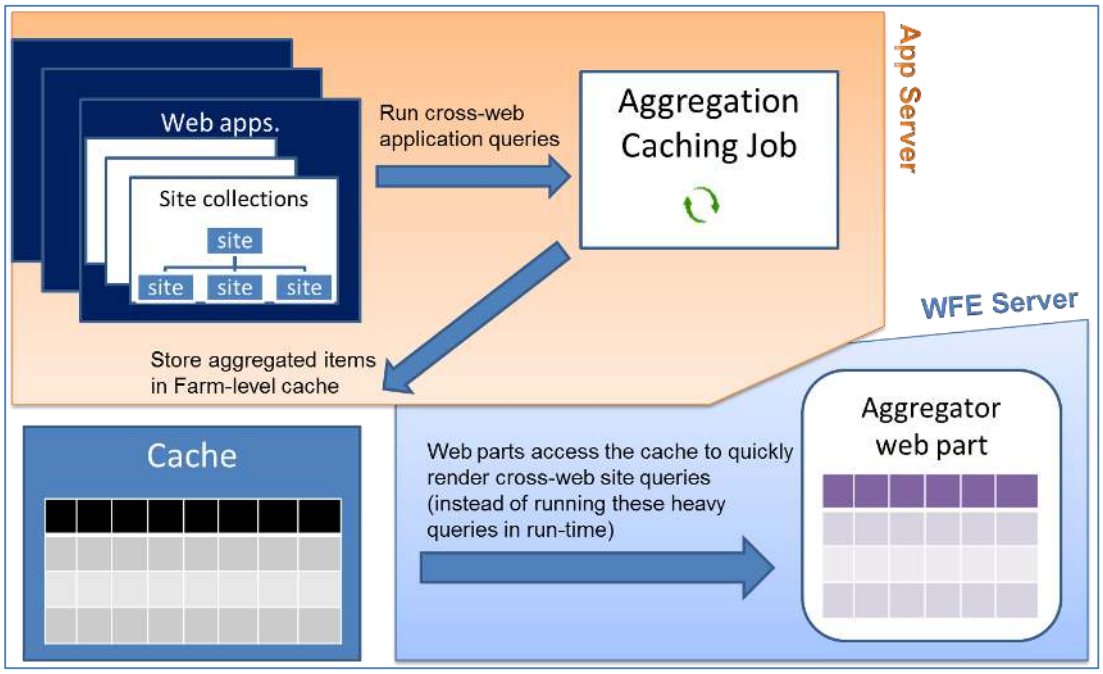
Application Programming Interface (API)

Date	Revision #	Revision	Author
April 6 2011	1	Document Created	Shai Petel
January 8 2014	2	Document reviewed	Inna Kerzman

KWizCom - Knowledge Worker Components
95 Mural St. Unit 600 Richmond Hill, Ontario L3R 9T3, Canada Phone +1-905-3700333
Email: info@kwizcom.com Web: http://www.kwizcom.com

Overview

KWizCom's *Enterprise Aggregation Caching Feature* is a server-based solution that enables to centrally manage and cache cross-web application aggregations.



This document describes the solution's API, which enables consumer software to use these aggregation caching services to display extensive aggregations' results in fast response time, with minimal load on your WFE servers.

Using DLLs

In order to use the aggregation caching API, you will need to add a using statement to:

1. SharePoint 2007: KWizCom.Utilities.dll
2. SharePoint 2010\2013: KWizCom.Foundation.dll

Access cache rule data

```
using (CacheUtility.ConfigWeb configWeb = new CacheUtility.ConfigWeb(false))
{
        try
        {
                SPList JobsList = configWeb.GetJobsList();
                SPListItem cacheItem = JobsList.GetItemById(RuleItemId);
```


KWizCom - Knowledge Worker Components
95 Mural St. Unit 600 Richmond Hill, Ontario L3R 9T3, Canada Phone +1-905-3700333
Email: info@kwizcom.com Web: http://www.kwizcom.com

```
                CacheUtility.CacheResultsData resultsData = null;
                byte[] tmp = configWeb.Web.GetFile(JobsList.RootFolder.ServerRelativeUrl + "/Attachments/"
+ cacheItem.ID + "/CacheXml.compressed").OpenBinary();
                tmp = KWizCom.Utilities.Utility.Compress.UnZip(tmp);
                string str = System.Text.Encoding.UTF8.GetString(tmp);
                resultsData =
KWizCom.Utilities.Utility.Serialization.XMLSerializeHelper.Deserialize<CacheUtility.CacheResultsData>(str,
System.Text.Encoding.UTF8);
                //Results are in this DataTable:
                resultsData.ResultsDataTable
        }
        catch { }
}
```

Working with the Aggregation Caching Rules list

This product was designed to enable users to work directly with the Aggregation Caching Rules list. As a developer you can integrate with that list directly to create new rules, modify existing rules and to request manual updates.

To do so, simply work with the SPList and SPListItem objects as you do with OOB SharePoint, and use the code sample above to get a handle of the configuration list.

Use ConstantsCaching class to get the list names, view names and all the field names in code.

CacheActionHandler.aspx

Description

This aspx page allows you to run various requests against the caching service, such as requesting manual update of a rule or getting a rules status report. Each call returns a JSON formatted string of an object with the rule status including tooltip and icon information.

Get cache rule data status

Run this in JavaScript, Requires jQuery to be enabled.
Note: This code was not optimised for production, it is provided for example purpose only. In production you should include this code in a namespace to avoid conflicts.

```
var lastPollTime = "";
function getRuleInfo() {
        var qs = "Action=GetInfo&Format=JSON&RuleID=" + selectedRuleID + "&CacheInvalidationKey=" + escape(lastPollTime) +
"&CacheBuster=" + new Date().getTime();
        $.get("/_layouts/EnterpriseAggregationCaching/CacheActionHandler.aspx?" + qs, getRuleInfoSucess);
}
function  getRuleInfoSucess(result) {
        var data = eval("(" + result + ")");
        var toolTip = data.Description;
        if (data.AllowQueueUpdateLink)
```


KWizCom - Knowledge Worker Components
95 Mural St. Unit 600 Richmond Hill, Ontario L3R 9T3, Canada Phone +1-905-3700333
Email: info@kwizcom.com Web: http://www.kwizcom.com

```
                ...
        else if (data.HasNewerData)
                ...
        else
                ...
        if (data.Status == "Ready")
                lastPollTime = data.CacheInvalidationKey;
        else if (data.Status != "Error") {
                //poll again in 10 seconds
                if (timeOutOperation != null)
                        window.clearTimeout(timeOutOperation);
                timeOutOperation = window.setTimeout(getRuleInfo, 10000);
        }
        ...
        ...
        ...
}
```

Request manual rule update

Run this in JavaScript, Requires jQuery to be enabled.

Note: This code was not optimised for production, it is provided for example purpose only. In production you should include this code in a namespace to avoid conflicts.

Note: requestUpdateSucess is optional, returns rule info and can be implemented the same as getRuleInfoSucess above.

```
function requestUpdate() {
        var qs = "Action=DoRefresh&Format=JSON&RuleID=" + selectedRuleID + "&CacheInvalidationKey=" + escape(lastPollTime) +
"&CacheBuster=" + new Date().getTime();
        $.get("/_layouts/EnterpriseAggregationCaching/CacheActionHandler.aspx?" + qs, requestUpdateSucess);
}
```

Implement Classes

ConfigWeb

Description:

Use this class to get the configuration web and to the aggregation rules list.

This class is disposable, and must be disposed at the end of your use or it will create a memory leak.

Code:

```
public class ConfigWeb : IDisposable
{
        public SPSite Site = null;
        public SPWeb Web = null;
        /// <summary>
        /// Returns Aggregator feature configuration web.
        /// Don't forget to dispose of this object!
        /// </summary>
        /// <param name="CheckIfFeatureIsActive">Only send true if you want an exception to be thrown if feature is not active. Otherwise
```


KWizCom - Knowledge Worker Components
95 Mural St. Unit 600 Richmond Hill, Ontario L3R 9T3, Canada Phone +1-905-3700333
Email: info@kwizcom.com Web: http://www.kwizcom.com

```
send false, to skip this test.</param>
        internal ConfigWeb()
        {
            SPSecurity.RunWithElevatedPrivileges(delegate()
            {
                SPFarm farm = SPFarm.Local;
                try
                {
                    Site = new SPSite(new Guid(farm.Properties[ConstantsCaching.AdminSiteId] as string));
                    Site.CatchAccessDeniedException = false;
                    Site.AllowUnsafeUpdates = true;
                    this.Web = Site.OpenWeb(new Guid(farm.Properties[ConstantsCaching.AdminWebId] as string));
                    this.Web.AllowUnsafeUpdates = true;
                }
                catch
                {
                    throw new Exception("Enterprise aggregation caching feature is not configured.");
                }
            });
        }
        public void Dispose()
        {
            if (Site != null)
                Site.Dispose();
            if (Web != null)
                Web.Dispose();
        }
        internal SPList GetJobsList()
        {
            SPList JobsList = null;
            try
            {
                var adminWeb = this.Web;
                JobsList = adminWeb.Lists[ConstantsCaching.JobListName];
                var provoke = JobsList.ItemCount;
            }
            catch (Exception ex)
            {
                throw new Exception("Failed to get " + ConstantsCaching.JobListName + " list on the selected site. Please
make sure this site is accesible and try again.", ex);
            }
                return JobsList;
        }
}
```

CacheResultsData

Description:

This class represents the deserialized cached results including:

DataTable, Field order, grouping definitions, collection of fields added to DataTable for programmatic needs (not to be displayed in the UI), and other information.

Code:

```
[Serializable]
public class CacheResultsData
{
```


KWizCom - Knowledge Worker Components
95 Mural St. Unit 600 Richmond Hill, Ontario L3R 9T3, Canada Phone +1-905-3700333

Email: info@kwizcom.com Web: http://www.kwizcom.com

```
string resultsDataTableXml = null;
public string ResultsDataTableXml
{
        set
        {
                resultsDataTableXml = value;
                resultsDataTable = null;
        }
        get
        {
                return resultsDataTableXml;
        }
}
string resultsDataTableXmlSchema = null;
public string ResultsDataTableXmlSchema
{
        set
        {
                resultsDataTableXmlSchema = value;
        }
        get
        {
                return resultsDataTableXmlSchema;
        }
}
DataTable resultsDataTable = null;
[XmlIgnore]
public DataTable ResultsDataTable
{
        get
        {
                if (resultsDataTable == null)
                {
                        try
                        {
                                resultsDataTable = new DataTable();
                                resultsDataTable.TableName = "AllResults";
                                using (StringReader sr = new StringReader(this.ResultsDataTableXmlSchema))
                                {
                                        resultsDataTable.ReadXmlSchema(sr);
                                }
                                using (StringReader sr = new StringReader(this.ResultsDataTableXml))
                                {
                                        resultsDataTable.ReadXml(sr);
                                }
                        }
                        catch { }
                }
                return resultsDataTable;
        }
}
/// <summary>
/// Marks all field that are needed to be added for grouping and other
/// view related tasks, even if they are not needed to display
/// </summary>
public List<string> FieldsToAdd = new List<string>();
/// <summary>
/// Marks all fields from FieldsToAdd that were added manually and
/// do not need to display
/// </summary>
public List<string> FieldsAddedManually = new List<string>()
/// <summary>
```


KWizCom - Knowledge Worker Components
95 Mural St. Unit 600 Richmond Hill, Ontario L3R 9T3, Canada Phone +1-905-3700333
Email: info@kwizcom.com Web: http://www.kwizcom.com

```
        /// Lookup fields added to view, but not added to query (need to add manually to the data table
        /// </summary>
        public List<string> FieldLookupFromView = new List<string>();
        /// <summary>
        /// key = [field name]
        /// value = [field caption]|true/false for isAsc order
        /// </summary>
        public DisctionarySerializeHelper.SerializableDictionary<string, string> GroupFieldsForCurrentView = new
DisctionarySerializeHelper.SerializableDictionary<string, string>();
        public DisctionarySerializeHelper.SerializableDictionary<string, string> SpecialFieldForCurrentView = new
DisctionarySerializeHelper.SerializableDictionary<string, string>();
                public DisctionarySerializeHelper.SerializableDictionary<string, int> FieldsOrdinal = new
DisctionarySerializeHelper.SerializableDictionary<string, int>();
}
```

ConstantsCaching

Description:

This class holds some constants values for the caching logic.

Code:

```
public class ConstantsCaching
{
        public const string AdminSiteId = "EAGCAdminSiteId";
        public const string AdminWebId = "EAGCAdminWebId";
        public const string JobListName = "KWizCom Aggregation Caching Rules";
        public const string JobList_View_Active = "Active Rules";
        public const string JobList_Field_ListType = "List type";
        public const string JobList_Field_SiteNames = "Site naming pattern";
        public const string JobList_Field_ListNames = "List naming pattern";
        public const string JobList_Field_AggViewUrl = "Aggregation view url";
        public const string JobList_Field_AggregatedSites = "Aggregated sites";
        public const string JobList_Field_AggUser = "Aggregation user account";
        public const string JobList_Field_Period = "Aggregation period";
        public const string JobList_Field_AllowRefresh = "Allow manual cache refresh";
        public const string JobList_Field_StartedRunning = "Started running";
        public const string JobList_Field_StartTime = "Start time";
        public const string JobList_Field_EndTime = "End time";
        public const string JobList_Field_Active = "Active";
        public const string JobList_Field_Status = "Job run status";
        public const string JobList_Field_StatusImg = "Status indicator";
        public const string JobList_Field_LastSuccessfulRun = "Last successful run";
        public const string JobList_Field_LastSuccessfulRunDuration = "Last run duration";
        public const string JobList_Field_ForceNextUpdate = "Queued for manual update";
}
```


KWizCom - Knowledge Worker Components
95 Mural St. Unit 600 Richmond Hill, Ontario L3R 9T3, Canada Phone +1-905-3700333

Email: info@kwizcom.com Web: http://www.kwizcom.com