# Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>August 11, 2016</td>
<td>Initial Draft</td>
<td>Support Team</td>
</tr>
<tr>
<td>1.1</td>
<td>August 16, 2016</td>
<td>Revised version with settingsitem overview</td>
<td>Support Team</td>
</tr>
</tbody>
</table>
# Contents

1. Objective .................................................................................................................. 3
2. Audience .................................................................................................................... 4
3. Product Description ................................................................................................... 5
4. Cireson Portal Components ....................................................................................... 6
   4.1 ServiceManagement Database ............................................................................. 6
   4.2 Cache Builder Service ......................................................................................... 6
   4.3 Web Site ............................................................................................................... 6
   4.4 Portal Communication Flow and Ports ................................................................. 6
5. Authentication Options ............................................................................................... 7
   5.1 Integrated Windows Authentication ................................................................. 7
   5.2 Forms Authentication ......................................................................................... 7
6. Installing The Cireson Portal ...................................................................................... 8
   6.1 Configure Prerequisites - Active Directory and Service Manager .................... 9
   6.2 Running the Installation .................................................................................... 10
7. Appendix 1 – Troubleshooting ............................................................................... 16
   7.1 Web Site Logging ............................................................................................. 16
   7.2 Re-syncing data from Service Manager to the Portal ....................................... 18
8. Appendix 2 - Portal View Descriptions and Criteria ................................................. 20
9. Appendix 3 – Cachebuilder Overview .................................................................... 22
   9.1 Data copied and cachebuilder schedules ....................................................... 22
   9.2 Cireson Cache Builder Log Files .................................................................... 23
10. Appendix 4 – Performance Tuning the Cireson Portal ............................................ 24
    10.1 Configuring Application Pool Recycling ......................................................... 24
    10.2 Configuring Application Initialization ............................................................ 24
    10.3 Configuring the Idle Timeout ....................................................................... 25
11. Appendix 5 – Administration Settings .................................................................. 26
    11.1 SettingsItem Overview ............................................................................... 28
12. Appendix 6 – Installing on a server without Internet Access .................................. 32
13. Appendix 7 – Tricks, Tips and additional Resources .............................................. 34
    13.1 Setting the popularity index on a request offering ....................................... 34
    13.2 Managing Enumerations for Knowledge Base ................................................ 35
    13.3 Additional resources ............................................................................... 38
1 Objective
This document is to provide end-to-end guidance and troubleshooting tips. It is intended to be used as a guide during the installation and setup of the Cireson free Cireson Self Service Portal solution. Existing customers installing the licensed version of the Cireson Analyst Portal should refer to documentation found the Cireson Support Portal.

For additional assistance, connect with Cireson customers, partners, experts and employees to learn, get answers to your questions and share new ideas on the Cireson Community:

https://community.cireson.com/categories/ssp-community-portal

NOTE:
Cireson Support (including phone & email) is not included with the free Self-Service Portal - Community app. Take advantage of the Cireson Community to find answers and for troubleshooting assistance. If you would like access to Cireson telephone, web, and email support, please contact us today.
2 Audience
This document is written for System Center administrators who are already familiar with the System Center Service Manager (SCSM) product and its functionality.

Any readers of this document should have a basic understanding of:

- Microsoft System Center Service Manager
- Basic SQL database maintenance processes
- Windows Server OS Technologies

Throughout this document, there are many items that deserve special attention and have been identified by the author as important or critical to the success of asset management. These are denoted in the following ways:

**NOTE:**

❗ Notes are used to draw attention to an important piece of information for the reader. Any detail contained within should be carefully read and understood.

**CRITICAL NOTE:**

❗❗ Critical notes are used to identify to readers a process step that is critical to success. Any detailed contained within should be completed before continuing.

**Cireson Best Practice:**

💡 Best Practice guidance will be shown wherever there is potential for implementation choices to impact the outcome. Each organization should consider this advice as it may not be right for every situation but instead should be taken as general advice.

**Code Example:**

💡 Any script or code example given will be enclosed to ensure simple copy and paste functionality if required.
3 Product Description

The Cireson Portal consists of three products the Self Service Portal, Analyst Portal and Dashboards.

The Self-Service Portal: The free community Cireson Portal for end users is a complete replacement of the Microsoft Self-Service Portal for Service Manager. Empower your end users to perform everyday self-service tasks by providing an easy to use, personalized experience when reporting issues, searching the knowledge base, and requesting services from the service catalog. In addition, the portal keeps customers informed of the status of requests and captures bi-directional communication between the service desk staff and the requestor. Reviewers can approve or reject requests and request implementers can update the status of their work. The Self Service Portal Includes the Cireson Knowledge Base and replaces the out-of-the-box Rich Text Format (RTF) knowledge base in Service Manager with a modern HTML offering. Provides users the information they need, when they need it, wherever they need it.

Analyst Portal: The Cireson Portal for analysts integrates seamlessly with Microsoft Service Manager to allow management of day-to-day activities on any browser, device, or OS. Built on cutting edge HTML 5, the Analyst Portal boasts fast, responsive, and highly functional capabilities. Items such as incidents, service requests, change requests, CMDB and reporting are all easily and quickly accessed and managed – with or without a desktop in sight. A valid Product Key is required to activate this functionality.

Dashboards: The Dashboards app for the Cireson Portal is an essential tool for any service management manager. Incident management, change management, and service request fulfillment dashboards provide insight into how a service management team is performing and helps identify hot spots and trends in the data. Managers can keep an eye on key performance indicators at a glance while they are on the go by accessing the dashboards through any browser. A valid Product Key is required to activate this functionality.

NOTE:

The Cireson Knowledge Base is independent of the Service Manager Knowledge Articles. All knowledge Base articles and list enumeration values are created and stored in the Cireson ServiceManagement Database. ServiceManager Knowledge Articles and list values are not accessible from the Cireson Knowledge Base.

NOTE:

The Advanced Request Offering functionality is only available in the Analyst Portal and will become active when a valid product key is entered.

A Product Key is designed for use with a specific environment and is required for any Licensed Cireson Application.
4 Cireson Portal Components
The Cireson Portal is comprised of three components - the ServiceManagement database, the Cache Builder, and the web site.

4.1 ServiceManagement Database
The ServiceManagement database is a standard SQL Server database that will be deployed automatically by the installer using a SQL Server "DacPac" (also known as a "Database Tier Application"). There only needs to be one ServiceManagement database per deployment of the Cireson Portal. All web sites can share a common ServiceManagement database.

4.2 Cache Builder Service
The Cache Builder service is a Windows service that is installed automatically by the installer. It is used to pull data out of Service Manager and Active Directory and store it in the ServiceManagement database as a cache of data designed to improve the performance of the Cireson Portal by storing the data in a format that is optimized for querying by the Cireson Portal. There only needs to be one Cache Builder service per deployment of the Cireson Portal. You can deploy multiple instances of the Cache Builder service and just have one of them running at a time.

4.3 Web Site
The Cireson Portal Web Site is an IIS web site that users connect to using a browser. There can be multiple web sites installed in a farm configuration behind a load balancer.

4.4 Portal Communication Flow and Ports
5 Authentication Options

When you install the portal, you will have a choice of using Integrated Windows Authentication (or what is sometimes called 'Single Sign On') or Forms Authentication. It is important to understand the differences between these authentication options prior to installation so that the correct choice is made at installation time. It is possible to change the authentication option after installation, but it is easier to consider the options and make the correct choice prior to installation.

5.1 Integrated Windows Authentication

Authentication method that allows the user of a Windows computer (not Linux or MacOS/iOS) to present his/her login credentials to the web server seamlessly to authenticate without being prompted. Integrated Windows Authentication uses the Kerberos Key Distribution Center which is part of an Active Directory environment to negotiate the authentication between the user’s browser and the server to attest that the user is in fact the person that is using the browser. When Integrated Windows Authentication is used, the Cireson Portal will impersonate the user to connect to System Center Service Manager. In order for Windows Integrated Authentication to work, the user’s computer, the user’s account, and the Cireson Portal web site web server must all be in the same domain or domains that have a trust relationship to each other. Typically, Integrated Windows Authentication is used in an intranet environment where the users are logged onto domain-joined Windows computers and connecting to a Cireson Portal web server that is in the same (or trusted) domain so that users do not have to spend time logging into the server. Cireson only supports using Negotiate (Kerberos, primary) and NTLM (fallback) for authentication providers and the app pool must use Kernel Mode Authentication. If you choose to deploy the Cireson Portal web site using Integrated Windows authentication, these settings will be set by default. If you choose to install using Integrated Windows Authentication, the Cireson Portal web site must be installed on a server that is also running a System Center Service Manager management server.

5.2 Forms Authentication

Authentication method that prompts the user to enter a username and password in a login form. The password and username are sent to the web server and authenticated by the web server using Active Directory as the authority. If the credentials are authenticated the user receives an authentication token which is stored in the browser and used to authenticate the user from that point forward for that particular session. If the session expires, the user will be required to reenter the username and password. The user has the option of checking the 'Remember me' checkbox on the login screen to securely store a login token in a cookie on the user’s local computer that can be used to authenticate instead of entering credentials for each session. The cookie will last as long as the configuration is set too last (default: 365 days) or until the user’s password changes and the authentication token is therefore no longer valid.

Once the user is authenticated by Active Directory, the Cireson Portal web site will create a connection to System Center Service Manager by securely passing the encrypted user name and password to the System Center Data Access Service on the Service Manager management server. The Data Access Service will then authenticate the credentials with Active Directory as well and if the credentials are valid and the user has permission to connect to Service Manager, a connection object will be created in the Cireson Portal web site and stored in memory. Because the username and password are sent on the network when using forms authentication,
it is essential to use HTTPS/SSL to encrypt the traffic on the network between the browser and the server so that login credentials are not transmitted in clear text. Forms authentication is typically used in a scenario where the users accessing the Cireson Portal web site are logged into non-Windows computers; using a browser other than IE or Chrome; are on Windows computers not joined to a domain where the Cireson Portal web server is located, or when users are accessing the Cireson Portal web site over the Internet.

6 Installing The Cireson Portal

The Cireson Portal must be installed on a server that has the following hardware and software requirements:

- System Center 2012 - Service Manager or System Center 2012 R2 - Service Manager (on the same server as the web site/cache builder)
- Windows Server 2012 or 2012 R2 (Windows Server 2008 R2 is not tested or supported)
- 8 GB of RAM minimum plus 100 MB per active, concurrent analyst user in a 24 hour period
- 4 cores minimum, 8 for larger deployments
- 1 GB of disk space

The ServiceManagement database must be installed on a server that has the following hardware and software requirements:

- 4 cores minimum, 8 for larger deployments (2.66 GHz or higher)
- 8 GB of RAM minimum
- 10 GB of disk space (with room to auto-grow as needed)
- SQL Server 2008 R2 SP1/SP2 or 2012 SP1/SP2 or 2012 R2. SQL Server 2014 is now supported. Standard or Enterprise versions
- Full text search enabled

Cireson Best Practice:

The Cireson Portal will perform best when the ServiceManager and ServiceManagement databases are located on the same SQL instance. The Cireson Portal is not supported on the same SQL instance as the Service Manager data warehouse or another SQL instance located on the same server.
6.1 Configure Prerequisites - Active Directory and Service Manager

Before you begin, you will need to do the following:

- **In Active Directory identify or create an Active Directory group** which represents all of your analysts.
- **In Active Directory identify or create an Active Directory group** which represents all of your knowledge managers (if you plan to use the HTML knowledge base app).
- **In Active Directory identify or create an Active Directory group** which represents all of your asset managers (if you plan to use asset management in the portal).
- **In Active Directory identify or create a user or service account** that the web site app pool identity will run as.
- **In Active Directory identify or create a user or service account** that the Cache Builder service will run as.
- **Create an Active Directory connector** in Service Manager that syncs in all of the above groups and users including all of the users contained in those groups.
- **Add** the **web site app pool identity and Cache Builder service account** to Service Manager in the **Administrator user role**.
- **Add** the **web site app pool identity and Cache Builder service account** as a **Local Administrator on the Portal Server**.
- **Add** the **Cache Builder service account as a dbo** on the **ServiceManager database**.

**CRITICAL NOTE:**

If creating new Active Directory groups for Analysts, KB Managers and Asset Managers ensure these groups have been synced into SCSM before continuing with Installation to avoid issues post installation as outlined [here](#).
6.2 Running the Installation

To begin installation login to a web server where you want to install the Cireson Portal web site as a user that is (a) a local administrator and (b) a SCSM administrator.

- Navigate to https://support.cireson.com/software in a browser on your web server.
- Click on the link or button of the version of the software that you want to install.
- If you cannot reach the URL via a browser, see the appendix 6 on how to download the files on an Internet-connected computer and then copy the files to the web server.
- On the home screen of the installer, click Let’s get started!
On the first page of the installer, choose to install all components or specific components:

Keep in mind that the database only needs to be installed once per deployment of the Cireson Portal.

The web site must always be installed.

The Cache Builder can optionally be installed, but there must be at least one Cache Builder per Cireson Portal deployment. Choose the language settings you prefer:

Click Next.
If you are installing the ServiceManagement database you will be prompted for the SQL Server name where the database should be installed and what you want to name your database.

**NOTE:**

⚠️ If the next button is greyed out, and there are no visible warnings or errors, please expand the advanced sections and verify all the information is correct. Once all the information is validated, the next button will become active.

**CRITICAL NOTE:**

Make sure that the logged in user that you are running the installer as is a sysadmin on the SQL Server where the ServiceManagement database will be installed. You can also choose to rename the database to something other than the default 'ServiceManagement' database name. If you click 'Advanced Database Settings' you can specify the location of the data (.mdf) and log (.log) files should be placed. These are local paths on the SQL Server not the server where the installer is running. If no path is specified the data and log files will be installed to the default directories for that SQL Server. You can specify a SQL Server instance name by entering the instance name in the textbox in the format of servername\instancename. For example mysqlserver\myinstance.
Enter the web site settings as follows:

- **Application Title** - this is the value that will be displayed on the login screen (if forms authentication is being used) and in the upper left of the header.

- **Service Manager Management Server Name** - If you are using Integrated Windows Authentication, the management server name must be localhost. If you are using Forms Authentication and the management server is installed on the computer where you are installing the Cireson Portal web site, then enter 'localhost'. If you are using Forms Authentication and the management server you want to connect to is not on the same server as where you are installing the Cireson Portal web site, enter the fully qualified domain name of the Service Manager management server that you want to connect to remotely. Note: connecting to a remote SCSM management server is not recommended.

- **Application Pool Identity** - the user or service account that you want to use for the application pool for the web site. This account should be a Service Manager administrator and a local administrator. Enter the username in the format domain\username.

- **Application Pool Identity Password** - enter the password for the Application Pool Identity. In some cases, the installer may not be able to verify the credentials provided and will show a warning. If you are certain that you have entered the correct credentials, you may proceed.

- **Notification Template Prefix** - this setting is used to filter out the list of notification templates which should be shown on the Send Email dialog in the Cireson Portal. Only those notification templates which have a display name that starts with the specified prefix value will be shown.

- **Return Email Address** - this is the email address that will be set as the return email address of any email sent via the Cireson Portal.

- **SMTP Server Name** - the name of the SMTP server to connect to when sending email.

- **SMTP Server Port** - the port number to be used.

- **Management Server SDK Path** - if you are installing the website on a non-management server you will need to provide either a UNC path to the SDK Binaries (i.e. \scsm\c$\Program Files\Microsoft System Center 2012 R2\Service Manager\SDK Binaries\) OR a local directory that you have manually copied the SDK Binaries to i.e. (C:\temp\SDK Binaries\). Once the installer runs the UNC can be removed.

- **(Advanced) Installation Directory** - the directory on the local file system to install to.

- **(Advanced) Web Site Name** - the name by which the web site will be known in IIS. This value is not shown to users.

- **(Advanced) Web Site Port** - the port number of the web site.

Click Next.
On the Cache Builder settings dialog enter the settings as follows:

- **Service Account** - this is the account that the Cireson Cache Builder Windows service will run as. Enter the name in the format domain\username. A domain account must be used. The domain account must be a local admin and a dbo on the ServiceManager database.

- **Service Account Password** - the password for the Service Account.

- **Analyst Group Name** - the name of the Group in AD which contains all of the analysts. This group must be in the same domain as the Service Account user. Enter just the group name. It is not necessary to specify the domain name.

- **Knowledge Base Managers Group Name** - the name of the Group in AD which contains all of the knowledge base managers. Same requirements as the Analysts group.

- **Asset Management Group Name** - the name of the Group in AD which contains all of the asset managers. Same requirements as the Analysts group. The members of this group will see the out of the box asset management views and will be able to create new asset management objects on the New menu.

- **Service Manager Management Server Name** - the name of the SCSM management server you want the cache builder to connect. This should always be 'localhost' because the Cireson Cache Builder must always be installed on a server that also is hosting a System Center Service Manager management server.
- **SCSM Database Server Name** - the name of the database server where the Service Manager database is.
- **SCSM Database Name** - the name of the ServiceManager database. (If you are connecting to an Always On Availability Group please see this guide)
- **(Advanced) Configuration Item Classes** - a comma separated list of CI class names if you want to limit the classes of CI objects that are sync’d into the cache.

Click Next. Accept the End User License Agreement and click Install.

If the installation fails or there are warning check the error and warning messages in the installer and follow the instructions provided if there are instructions. Click the View Log if there are issues and review the notes in the log.

If the installer has a warning message that the management pack was not successfully imported there could be many problems for this (insufficient permissions, lack of connectivity to SCSM, SCSM data access service is not running, etc.). The easiest way to workaround this is to just import the management packs manually in SCSM. The management pack files can be found by running the installer again and clicking the 'Browse installer files' on the home page and then navigating to the directory /InstallationFiles/ManagementPacks.
Appendix 1 – Troubleshooting

7.1 Web Site Logging
The Cireson Portal log is located at C:\inetpub\CiresonPortal\Logs\WebConsole.log.

The logging configuration is in the web.config file at C:\inetpub\CiresonPortal\web.config.

Changing the Web Site Logging Level
If you need to change the logging level to get more detail in the log, please do the following:

- Open a text editor such as Notepad as an elevated administrator.
- Open C:\inetpub\CiresonPortal\web.config.
- Change the logging level key to ALL. Example shown below:

```xml
<logger name="Logger">
  <!-- LEVELS: ALL, DEBUG, INFO, WARN, ERROR, FATAL-->
  <!-- Suggested for debugging: INFO -->
  <!-- Suggested for production: WARN -->
  <level value="ALL"/>
  <appender-ref ref="ConsoleAppender" />
  <appender-ref ref="RollingLogFileAppender" />
</logger>
```

- Save the file.
- Restart the Cireson Portal web site.

NOTE:
Once you no longer need the more detailed level of logging, please change the logging level back to ERROR to minimize the impact of debug logging on performance and disk space consumption.

Creating a New Web Site Log File
To create a new web site log file, please do the following:

- Stop the Cireson Portal web site.
- Delete or rename the current web site log file at C:\inetpub\CiresonPortal\Logs\WebConsole.log
- Start the Cireson Portal web site.

A new web site log file will be created automatically.
**Error Events**

All errors that occur in the Cireson Portal web site are also logged to the Application event log in Windows. The source of these events is 'WebPortal'.

**Cache Builder Logging**

The Cache Builder log file is located at C:\inetpub\CiresonPortal\bin\Logs.

The Cache Builder logging configuration is stored in the Cache Builder config file at C:\inetpub\CiresonPortal\bin\Cireson.CacheBuilder.WindowsService.exe.config.

**Changing the Cache Builder Logging Level**

If you need to change the logging level to get more detail in the log, please do the following:

- Open a text editor such as Notepad as an elevated administrator.
- Open C:\inetpub\CiresonPortal\bin\Cireson.CacheBuilder.WindowsService.exe.config.
- Change the logging level key to ALL. Example shown below:

```xml
<logger name="Logger">
  <!-- LEVELS: ALL, DEBUG, INFO, WARN, ERROR, FATAL -->
  <!-- Suggested for debugging: INFO -->
  <!-- Suggested for production: WARN -->
  <level value="ALL"/>
  <appender-ref ref="ConsoleAppender" />
  <appender-ref ref="RollingLogFileAppender" />
</logger>
```

- Save the file.
- Restart the Cache Builder service.

**NOTE:**

Once you no longer need the more detailed level of logging, please change the logging level back to ERROR to minimize the impact of debug logging on performance and disk space consumption.

**Creating a New Cache Builder Log File**

To create a new Cache Builder log file, please do the following:

- Stop the Cache Builder service. Wait for up to 15 seconds for the process to release control of the log file.
- Delete or rename the current Cache Builder log file at C:\inetpub\CiresonPortal\bin\logs
- Start the Cache Builder service
A new Cache Builder log file will be created automatically.

**Error Events**

All errors that occur in the Cache Builder are also logged to the Application event log in Windows. The source of these events is ‘CacheBuilder’.

### 7.2 Re-syncing data from Service Manager to the Portal

Sometimes for troubleshooting, it is recommended to re-sync data with the Cache Builder from SCSM to the Cireson ServiceManagement Database, please follow the steps below:

- Stop the cachebuilder service
- Run this SQL query against ServiceManagement: TRUNCATE TABLE LastModified
- Start the cachebuilder service

The Cache Builder used to completely rebuild everything anytime it was restarted, but now it just does updates based on timestamps found in the LastModified table. Keep in mind that this will not necessarily remove data that is present in Service Manager

**Completely Re-sync your Enumerations (list values)**

In order to completely delete all enumerations brought in by the cache and force a refresh, run this against your ServiceManagement db:

```
DELETE DisplayString WHERE ElementID IN (SELECT EnumerationID FROM Enumeration WHERE CreatedBy = 'c6745a66-5ccc-4fbc-b1d8-ab9797cdea2d');
DELETE Enumeration WHERE CreatedBy = 'c6745a66-5ccc-4fbc-b1d8-ab9797cdea2d';
TRUNCATE TABLE LastModified;
```

Then restart the cachebuilder.

**CRITICAL NOTE:**

If you have disabled any enumerations, this may re-enable them.

**Completely Re-sync your Service Offerings and Request Offerings**

In order to completely re-sync your Service Offerings and Request Offerings, run this against your ServiceManagement db:

- TRUNCATE TABLE LastModified;
- TRUNCATE TABLE ServiceOffering;
- TRUNCATE TABLE RequestOffering;

Then restart the cachebuilder.
If you are truncating these tables to ensure that your SO and ROs are up to date you may need to also restart the website and recycle the application pool following the truncate.

**CRITICAL NOTE:**

This will temporarily remove all Service and Request Offerings from the portal until the cachebuilder rebuilds them.

This will also remove the popularity settings from ROs if you have this set.

---

**How to Remove Work Items that have been Deleted from ServiceManager via PowerShell**

When removing workitems from Service Manager using powershell you will also need to clear the workitem record from the Cireson ServiceManagement database. There are a couple of options to remove deleted work items from the ServiceManagement db listed below.

Selectively delete work items from the cache. Replace IR123 with the WorkItemld of the work item you wish to remove:

- `DELETE WorkItem WHERE WorkItemId = 'IR123'`

Probably the easiest to remove all at once:

- `TRUNCATE TABLE LastModified;`
- `TRUNCATE TABLE WorkItem;`

Then restart the cachebuilder.

---

**How to Remove Users that have been Deleted from the ServiceManager Database**

First stop the Cache Builder service and then run the following commands against the ServiceManagement DB:

- `TRUNCATE TABLE CI$User`
- `TRUNCATE TABLE CI$DomainGroup`
- `TRUNCATE TABLE LastModified`

Then restart the CacheBuilder service.
## Appendix 2 - Portal View Descriptions and Criteria

<table>
<thead>
<tr>
<th>View</th>
<th>App</th>
<th>Criteria</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>My Requests</td>
<td>Self-service Portal</td>
<td>All work items where the logged in user is related to the work item as the affected user.</td>
<td>This Active tab on this view shows only work items where the status is not closed. The Closed tab shows only work items where the status is closed.</td>
</tr>
<tr>
<td>Team Requests</td>
<td>Self-service Portal</td>
<td>All work items where the logged in user is related to the work item as the affected user or the created user PLUS work items where the affected user is in the same &quot;team group&quot; as the logged in user.</td>
<td>Team groups can be configured in the Portal Admin settings.</td>
</tr>
<tr>
<td>My Work</td>
<td>Self-service Portal and Analyst Portal</td>
<td>All work items where the logged in user is related to the work item as the assigned to user or primary owner PLUS All work items where the logged in user is a member of a group (recursively) that is related to the work item as the assigned to user or primary owner PLUS all review activities where the logged in user is a reviewer PLUS all review activities where the logged in user is a member of an AD group (recursively) that is a reviewer</td>
<td></td>
</tr>
<tr>
<td><strong>Team Work</strong></td>
<td><strong>Analyst Portal</strong></td>
<td>All work items where the logged in user is related to the work item as the assigned to user PLUS all review activities where the logged in user is a reviewer. Note: the user must be a reviewer of a review activity directly and not just be a member of a group that is a reviewer. PLUS all work items where the support group property is mapped to an Active Directory group that the user is a member of.</td>
<td>Support group mapping can be configured in the SCSM console Administration\Settings view.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Active Work</strong></td>
<td><strong>Analyst Portal</strong></td>
<td>All work items where the status does not equal Closed.</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 3 – Cachebuilder Overview

The Cireson Cache Builder retrieves information from the ServiceManager database, transforms it (for example; flattens hierarchies and merges data from multiple tables into one), and then stores it in the Cireson ServiceManagement database.

The following permissions are required for the Cache Builder service account:

- Must be a member of the SCSM Administrator user role
- sysadmin rights to the ServiceManager and ServiceManagement databases

The following ports should be available to the Cache Builder service:

- 5724 - For the System Center Data Access Service on the SCSM management server
- 1433 - Default port used to communicate with SQL Server

Data copied and cachebuilder schedules

The schedules are based on the time when the Cache Builder was last restarted.

For example, if the service was started at 8:05 a.m. then the Scoped Access operation would run immediately, and then again at 8:05 a.m. the following day and every day thereafter.

<table>
<thead>
<tr>
<th>Area</th>
<th>Default Schedule</th>
<th>Operation Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Items</td>
<td>On startup and once per minute thereafter</td>
<td>Incremental</td>
<td>Work item scoped access is also updated if work items are added/removed from queues that are used by user roles for scoping or users are added/removed from user roles.</td>
</tr>
<tr>
<td>Configuration Items</td>
<td>On startup and every hour</td>
<td>Incremental</td>
<td>Website restart is required to pick up changes to this type of data because it is cached in the web site memory at web site startup.</td>
</tr>
<tr>
<td>Users and Groups</td>
<td>On startup and every two hours</td>
<td>Incremental</td>
<td>Website restart is required to pick up changes to this type of data because it is cached in the web site memory at web site startup.</td>
</tr>
<tr>
<td>Service Catalog</td>
<td>On startup and every 24 hours</td>
<td>Incremental</td>
<td></td>
</tr>
<tr>
<td>Enumerations (List Values)</td>
<td>On startup and every 24 hours</td>
<td>Incremental</td>
<td>Website restart is required to pick up changes to this type of data because it is cached in the web site memory at web site startup.</td>
</tr>
<tr>
<td>Scoped Access</td>
<td>On startup and every 24 hours</td>
<td>Complete rebuild</td>
<td>Website restart may be required to pick up changes to this area. This operation can take quite a while so it is recommended that you schedule this to run during off-peak hours. See below as to what is encompassed.</td>
</tr>
</tbody>
</table>
Scoped access is a broad area that encompasses the following:

1. **User access to:**
   Work items, Config items, Service catalog, Various “views” found on the portal

2. **Determining and storing group membership for the following groups**
   Analyst, AssetManager, and KnowledgeManager groups, Team groups, Support groups, Form groups, Security groups, Groups targeted by folders, sections, and links on the Navigation Admin page, Groups targeted by announcements

Currently, all groups listed above must be located in the same domain as the Cache Builder service account (but may contain users from other domains/forests).

Group membership is determined by communicating with Active Directory – this is why this operation is not incremental, we don’t currently have a way to retrieve changes made to Active Directory in a given time frame.

**Changing Default Cachebuilder schedules**

- Disable and stop the Cireson Cache Builder service and terminate any Cireson.CacheBuilder.WindowsService.exe processes in the task manager, if needed.
- Open notepad or a similar text editor as an administrator and then open the file at C:\inetpub\CiresonPortal\bin\Cireson.CacheBuilder.WindowsService.exe.config (that’s the default path, yours may be different).
- Look for the “cacheCommandSection” and you’ll see a section that looks like:
- Edit the refreshInterval (measured in minutes) for each operation as desired. Keep in mind that running cache updates too frequently can have an overall negative impact on performance.
- Save the config file
- Enable and start the Cireson Cache Builder service

9.2 **Cireson Cache Builder Log Files**

The Cireson Cache Builder service log is located at C:\inetpub\CiresonPortal\bin\logs.
10 Appendix 4 – Performance Tuning the Cireson Portal

10.1 Configuring Application Pool Recycling
A slow page load time can indicate that the application pool has been recycled since the last time the portal was accessed. When a user accesses a page on the Cireson Portal for the first time after an application pool recycle, an application start event occurs which will pull the ServiceManagement database cache into the browser memory cache. Especially for large customer environments this process can take up to a few minutes. This is a one-time event that occurs after each application pool recycle. Further, for each page that is accessed for the first time after an application pool recycle, IIS will take a little longer to generate the page and store the resulting page in the IIS output caching. This needs to be done only once for each page. Once the web site memory cache is populated and a page has been cached in the output cache, subsequent requests for that page should be fast. The application pool recycles every 29 hours by default, so users can experience this as a slow initial load on a near-daily frequency. Because the default is 29 hours it will occur at a different time each day, sometimes occurring during business hours. Also, administrators may be recycling more often, especially if they are working on configuring the portal for an initial deployment.

One thing that can be done to help address this is to adjust application pool recycling frequency/conditions. While Cireson does not recommend shutting off app pool recycling completely, it can be delayed to occur once every few days or during specific time windows. To do this, open IIS:

- Navigate to Application Pools.
- Right click on CiresonPortal.
- Select Recycling...
- Edit the Recycling conditions. The default setting is 1740 minutes (29 hours). Cireson recommends only having the app pool recycle automatically 1/day during non-business hours.

10.2 Configuring Application Initialization
One thing you can do to minimize the performance impact of the application start event and the process of loading cache data from ServiceManagement into the web site memory cache is to immediately execute the app start event code after an application pool recycle instead of waiting for the first user to request a page from the web site. This can be done by configuring a feature of IIS called ‘Application Initialization’.

To use Application Initialization, you must first install the feature using the Add Roles/Features wizard in the Server Manager (not Service Manager) console.
To configure Application Initialization, do the following:

- Open IIS
- Navigate to Application Pools in the navigation tree on the left
- Right click on the CiresonPortal application pool and choose Advanced Options
- Change Start Mode to AlwaysRunning
- Navigate to the Cireson Portal web site on the left navigation tree.
- Double click 'Configuration Editor' in the center panel bottom row.
- In the drop down at the top, select system.webServer/applicationInitialization
- Change the doAppInitAfterRestart to true
- Click Apply on the right

Now, when you restart the web site a background process will try to access a page on the site immediately after the application pool is restarted (instead of waiting for the first user to access the page and trigger the cache load application start event). If you wait for a few minutes after a web site restart and then access the site you should see that the first page access time is minimal because the cache has already been populated in the background.

10.3 Configuring the Idle Timeout

By default, when an IIS web site is installed it is configured to shut down the application pool if it is idle for 20 minutes. You can change the Idle Time-Out value in settings like this:

- Open IIS
- Navigate to Application Pools in the navigation tree on the left
- Right click on the CiresonPortal application pool and choose Advanced Options
- Change Idle Time-Out (minutes) to whatever value you want. Cireson recommends 0 (don’t ever stop the application pool due to inactivity) for a production environment and 60 minutes for a non-production environment.
11 Appendix 5 – Administration Settings

This section describes the options found in the Administrator Settings.

**License Key**

The first Administrator Settings are the Product key. When a valid product key is provided, the corresponding functionality will become available in the portal.

**License Key Functionality**

*Analyst Portal: My Work, Team Work, Active Work, Search (including creating persisted search views), Configuration Items and RTF Knowledge Base view, New drawer menu for creating new work items,*

*Dashboards: Dashboards views.*

**Work Item Grid Refresh Interval**

The Work Item grid refresh interval controls how frequently the work item grid views automatically refresh. Keep in mind that many users having work item grid views open with a frequent refresh interval will have a potential impact on the ServiceManagement database. Generally speaking, the larger the number of users the longer the refresh interval should be. A best practice is to start with a 5-minute refresh interval and change it to refresh more frequently and measure the impact on database performance and user experience.

**Create New Work Items**

These settings control the visibility of the New menu work item options. The New menu work item options are only shown if the Analyst Portal product key is entered and valid. By default all users that are members of the Analysts group entered during setup can see the New menu work item options. If the checkbox for a given work item type is unchecked, the New menu options will not be displayed for any user. If the checkbox is checked and a group is specified only members of that group will see the New menu options for that work item type. Normally, any analysts are allowed to create any type of work item and therefore the default configuration of the work item type checkboxes being checked and no groups being specified is sufficient because by default any analyst can create any type of work item. Specifying a group is only
useful to limit creating a certain type of work item to a subset of users in the Analysts group. In addition to an analyst user being granted permission to view the work item options in the New menu, the analyst user must also be included in a SCSM user role which grants the user permission to create work items.

A typical configuration would be to do the following:

- Create an "Analysts" Active Directory group which contained all of the analyst users. The group can contain users or other groups.
- Add the "Analysts" Active Directory group to the Advanced Operators SCSM user role. Alternatively, you could add the "Analysts" group to the following user roles: Incident Resolvers, Problem Analysts, Service Request Analysts, Change Managers, Activity Implementers, Release Managers.
- Specify the "Analysts" group during the setup of the Cireson Portal.
- Leave each of the Create New Work Items checkboxes checked. Do NOT specify a group for each of the work item types.

Note: the AD group (and any subgroups) must belong to the same domain as the cachebuilder service account, but may contain users from another domain.

Activity Settings

The activity settings control permissions to perform certain tasks on review activities and manual activities. By default, all users have permissions to perform these tasks. Specifying a group for these settings limits who can perform these tasks.

Manual Activity Settings

- Mark as Completed/Failed controls the visibility of those statuses in the status dropdown of a manual activity.
- Edit Activity controls whether or not the manual activity form fields are enabled or disabled.

Review Activity Settings

- Approve/Reject and Add/Edit/Remove Reviewer controls who can see the buttons for performing those actions on the review activity form.
- Edit Activity controls whether or not the manual activity form fields are enabled or disabled.

Assign Forms to Active Directory Groups

This section allows an administrator to target custom forms to groups of users in Active Directory so that depending on a user’s membership in an Active Directory group and its assignment to a custom form, different users can see different forms for a given work item type. For example, you may want to present a form to HR service request analysts which shows some custom properties of the service request class which are specific only to HR service requests. Another example might be to make some fields only editable for certain users.
When creating a form assignment, you need to enter the following:

- **Active Directory group name.** This group must exist in the ServiceManagement database (the group, and any subgroups, must belong to the same domain as the cachebuilder service account, but may contain users from another domain). You can verify that it is present by running this query: `SELECT * FROM CI$DomainGroup WHERE Username = 'the group name'`

- **The ID of the custom form as defined in the work item .js file.** For more information on creating custom forms, search the KB for 'custom forms'. Note: form names are case-sensitive.

- **The ordinal which is an integer that controls the order in which a form assignment is applied.** For example, a user may be a member of more than one Active Directory group. Each of those groups may be assigned a different incident form. The form that is assigned with the lowest ordinal will be applied.

- **Form type - the type of form - incident, service request, etc.**

- **The form projection ID.** The ID of the SCSM type projection that defines the shape of the data (seed object class, properties, and related object classes) that will be pulled to populate data on the custom form and save the data back to SCSM on save/apply.

Assigning a user group, a custom form for a given work item type does not grant users that are members of that group to create, update, or delete data in SCSM using that form. The create, update, delete permissions are still controlled by SCSM user roles.

**Team Requests View Groups**

The Team Requests view is a special view provided out of the box that shows work items where the logged in user is the affected user and work items where the affected user is a member of any of the same specified 'Team Groups' as the logged in user. For example, if Bob and Joe are both engineers in the Network Engineering Department, a group could be created in Active Directory and contain Bob and Joe. That group could then be added to the Team Requests View Groups in the Cireson Portal administrator settings. When Bob logs in to the portal and navigates to the Team Requests view, he will see work items where he is related to the work item as the affected user and also where Joe is related to the work item as the affected user. The Team Requests view is also useful for scenarios where a service provider is supporting multiple customers and each of the end users in each of those customers’ needs to be able to view the work items created by people in their own company but not others.

**Important notes:**

- The Analysts group specified during setup should not be used as a Team Requests view group.

- These groups (and any subgroups) must belong to the same domain as the cachebuilder service account, but may contain users from another domain.

**11.1 SettingsItem Overview**

Settings entered during the installation can be updated or changed in the settingsitem table.
<table>
<thead>
<tr>
<th>Key</th>
<th>Setting Explanation</th>
<th>Applies to Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalystHomePage</td>
<td>Landing page when Analysts first log into portal</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>AnalystPortalKey</td>
<td>License key for Analyst Portal to activate Analyst features of the portal</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>AnalystsADGroup</td>
<td>Username of Active Directory group to define which users have Analyst Level Access</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ApplicationTitle</td>
<td>Name of application displayed top left hand corner and on login page if using forms authentication</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>AssetManagementConsumableTemplate</td>
<td>Default Asset Management Consumable Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>AssetManagementContractLeaseTemplate</td>
<td>Default Asset Management Contract Lease Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>AssetManagementInvoiceTemplate</td>
<td>Default Asset Management Invoice Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>AssetManagerADGroup</td>
<td>Username of Active Directory group to define which users have access to Asset Objects, Views and Tasks</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>AutoRefreshLicenses</td>
<td>Sets if licenses will be automatically retrieved from licensing.cireson.com</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>CacheBuilderMaxRetries</td>
<td>This setting is used to retry a synchronizing command within cachebuilder, should a command fail the cachebuilder will retry this number of times before failing permanently.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>CacheBuilderRetryInterval</td>
<td>See cachebuilder overview</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ConfigItemClasses</td>
<td>Comma separated list of CI class names if you want to limit the classes of CI objects that are sync’d into the cache.</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ContractWarrantyTemplate</td>
<td>Default Contract Warranty Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>CreateOnBehalfGroup</td>
<td>Username of Active Directory group for which members will see the Create On Behalf Of control on the Request Offering Form.</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>CRGroup</td>
<td>Group of users who can create Change Requests in the drawer (Null = all Analyst Users)</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>CustomizableLoginLink</td>
<td>Text to be displayed for the Custom link on forms login page.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>CustomizableLoginLinkUrl</td>
<td>Custom login link URL for forms login page</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>DashboardKey</td>
<td>Auto populated or manually entered value to activate the dashboards menu</td>
<td>Analyst Portal, Dashboards</td>
</tr>
<tr>
<td>DefaultIncidentTemplate</td>
<td>Default template for quick create and new workitem task</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>DefaultProblemTemplate</td>
<td>Default template for quick create and new workitem task</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>DefaultReleaseRecordTemplate</td>
<td>Default template for quick create and new workitem task</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>DefaultServiceRequestTemplate</td>
<td>Default template for quick create and new workitem task</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>DontUseScopedAccess</td>
<td>True/False to respect user role scoping to workitems and request offerings</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>EndUserHomePage</td>
<td>Landing page when End users first log into portal</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>EnvironmentHashCode</td>
<td>Unique environment ID used when to retrieve license key from product key</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ForceSearch</td>
<td>Hide request offerings so users must first search for what they want before to see desired offerings</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>GridViewRecordCap</td>
<td>Capped limit and maximum number of results that will be displayed in a query result prompt in a request offering form.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>GroupQuery</td>
<td>Query criteria for pickers to find group objects, display and username by default</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>HardwareAssetTemplate</td>
<td>Default Hardware Asset Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>IRGroup</td>
<td>Group of users who can create Incident Requests in the drawer (Null = all Analyst Users)</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
<td>Portal(s)</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>KbSearchByUserLocale</td>
<td>When set to True any Knowledge Base searches will only return articles that are set to the user’s preferred language.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>KnowledgeManagerADGroup</td>
<td>Username of Active Directory group to define which users can create and edit KB Articles</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>LanguageCode</td>
<td>Localization code for language of Portal</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>LanguageCodeDateTime</td>
<td>Language Code to show correct Date formats of date properties</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>LastModifiedAdjustmentInMilliseconds</td>
<td>Workflows and other processes may complete an update sometime after the time specified in the Last Modified field. This gap can be accounted for by the LastModifiedAdjustmentInMilliseconds.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>LocalizationManagerADGroup</td>
<td>AD Group to define which users can over-ride localization strings</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>MACompleted</td>
<td>Group of users who can complete Manual Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>MAEdit</td>
<td>Group of users who can edit Manual Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>MAFailed</td>
<td>Group of users who can fail Manual Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>ManagementServer</td>
<td>SCSM management server name - use localhost when Portal and Management Server on the same machine.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>NewCR</td>
<td>Shows or hides New Change Request button depending on value</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>NewIR</td>
<td>Shows or hides New Incident Request button depending on value</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>NewPR</td>
<td>Shows or hides New Problem Request button depending on value</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>NewRR</td>
<td>Shows or hides New Release Request button depending on value</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>NewSR</td>
<td>Shows or hides New Service Request button depending on value</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>NoticeEventTemplate</td>
<td>Default Notice Event Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>NotificationTemplatePrefix</td>
<td>Template for the send email task of the portal</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>PrePopulateCiLimit</td>
<td>The value of this setting is an integer with a default of 1. Meaning any workitem that has at least 1 related Configuration Item will perform the query to prepopulate before the save occurs. For environments with large amounts of CI classes this number can be increased to something more appropriate such as 10, in this case the prepopulate would only occur if the workitem being saved had more than 10 related CIs</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>PRGroup</td>
<td>Group of users who can create Problem Requests in the drawer (Null = all Analyst Users)</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ProductKey</td>
<td>Product key for your environment to activate the licensed products you have purchased</td>
<td>Analyst Portal, Dashboards</td>
</tr>
<tr>
<td>PromoteViewConnectionString</td>
<td>Server URL connection for promoting console views into the portal</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>PurchaseOrderTemplate</td>
<td>Default Purchase Order Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>PurchaseTemplate</td>
<td>Default Purchase Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>RAAddReviewer</td>
<td>Group of users who can add reviewers to Review Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RAApprove</td>
<td>Group of users who can approve Review Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RADeleteReviewer</td>
<td>Group of users who can delete reviewers from Review Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RAEdit</td>
<td>Group of users who can edit Review Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RAEditReviewer</td>
<td>Group of users who can edit reviewers on Review Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RAReject</td>
<td>Group of users who can reject Review Activities (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>RefreshIntervalMilliseconds</td>
<td>This determines how often a grid will be refreshed with data.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RememberMeExpirationDays</td>
<td>After this point the user will be required to reenter the credentials using forms authentication and can optionally choose Remember Me again to create a new token</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RightToLeft</td>
<td>This setting should be set to true when utilizing the portal with a RightToLeft language option.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RRGroup</td>
<td>Group of users who can create Release Requests in the drawer (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>RunConcurrencyCheck</td>
<td>When set to True a pre-check will be done before saving a workitem to ensure no changes have been made in the interim of when the user loaded the workitem and when they click to save it.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>SearchGroupsInUserPicker</td>
<td>Have group objects hidden or shown in the user picker depending on value</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>SearchUsersInUserPicker</td>
<td>Have user objects hidden or shown in the user picker depending on value</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>ServiceCatalogLanguageCode</td>
<td>Wildcard by default, shows all service offering language codes on service catalog</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>ServiceManagerInactivityTimeoutMinutes</td>
<td>Timeout duration of the token cookie</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>ShowNewButtonInDrawer</td>
<td>Show drawer button to create objects</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ShowQuickCreateForIR</td>
<td>Show or hide quick create for Incidents Requests on the drawer</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ShowQuickCreateForPR</td>
<td>Show or hide quick create for Problem Requests on the drawer</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ShowQuickCreateForRR</td>
<td>Show or hide quick create for Release Records on the drawer</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>ShowQuickCreateForSR</td>
<td>Show or hide quick create for Service Requests on the drawer</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>SMTPEmailReturnAddress</td>
<td>From address when using send email task in the portal</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>SMTPServerName</td>
<td>Server name for send email task in the portal</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>SMTPServerPort</td>
<td>Server port for send email task in the portal</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>SoftwareAssetTemplate</td>
<td>Default Software Asset Template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>SRGroup</td>
<td>Group of users who can create Service Requests in the drawer (Null = all Analyst Users)</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>SupportandMaintenanceContractTemplate</td>
<td>Default Support and Maintenance Contract template</td>
<td>Analyst Portal</td>
</tr>
<tr>
<td>TeamGroupFilter</td>
<td>This should be set to True when the client has configured Team Groups within Admin Settings and wishes to limit the access to workitems and User lists according to team group membership.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>UpdateUserProfileNagInterval</td>
<td>The setting is the number of days that is subtracted from the User Profile Last Reviewed Date to determine whether to notify the user to review their profile.</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>UseNewMultipleObjectControl</td>
<td>TBD ... currently in development</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>UserQuery</td>
<td>Query criteria for pickers to find user objects, can be changed to add ability to search EmployeeID etc</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>UserTokenExpirationMinutes</td>
<td>Inactivity timeout in minutes</td>
<td>Self Service Portal, Analyst Portal</td>
</tr>
<tr>
<td>VendorTemplate</td>
<td>Default Vendor Template</td>
<td>Analyst Portal</td>
</tr>
</tbody>
</table>
Appendix 6 – Installing on a server without Internet Access

Installing the latest version of the Cireson Portal on a server without internet access can be achieved following these steps:

1. On your local computer that has Internet Access go to the Software page on the Cireson Support Portal:  https://support.cireson.com/software/

2. Click the Install button

3. In the case of using Internet Explorer the Setup Application Installation will begin automatically it may ask you to allow the app to make changes on your computer. After clicking ‘Yes’ you will arrive at the Portal Installer Welcome Screen and follow on to Step 4.

In the case of using Firefox or Chrome you may be prompted to save the Application Manifest file:

Once saved open the file on your local computer and select Install (this will NOT install the Portal just securely download the Portal Installer files):
Once the Setup Application Installation completes and you arrive at the Portal Installer Welcome Screen.

4. Click the “Browse installer files” link in the lower right corner of the window:

   ![Browse installer files](image)

5. This will bring you to a local file folder, copy the entire folder contents including all subfolders and place it on the server you wish to install to. At this point you can close/cancel the Portal Installer.

   ![Cireson Setup.exe](image)

6. Once the files are copied over log into the server and move to the folder where you saved the installation files and double-click the Cireson Setup.exe to begin your installation.
13 Appendix 7 – Tricks, Tips and additional Resources

13.1 Setting the popularity index on a request offering
Currently the only way to set the popularity of a Request Offering is changing the data in the database directly. Below is the information needed in order to accomplish this as well as what effect it has to the end user.

**What the popularity index does**

Setting the popularity index will impact what the user sees on the home page of the Portal. Listed under *Top Requests* you will see Request offerings sorted by popularity. The lower the number the higher the popularity. So if you have 5 Request Offerings and you set the popularity to 10, 20, 30, 40, and 50 they will be listed starting with the Request Offering that is set to 10 and ending with the Request Offering that is set to 50.

**How to update the popularity index**

You will need access to the database the Portal is running on. The target information is

- Database: ServiceManagement (unless changed from default)
- Table: RequestOffering
- Column: Popularity
- DataType: Integer (nulls allowed)

Open a new query window in SQL Management Studio. Make sure the target database is correct. Adjust the query below to contain the proper where clause for your filtering.

UPDATE [dbo].[RequestOffering] SET [Popularity] = {{POPULARITY}} WHERE {{CRITERIA}}

**Example 1: Setting popularity using ID as the selection criteria** -

UPDATE [dbo].[RequestOffering] SET [Popularity] = 42 WHERE ID = ‘C90D9ABC-E71D-8A34-EAE7-20E63F278C47’

**Example 2: Setting popularity using Title as the selection criteria** -

UPDATE [dbo].[RequestOffering] SET [Popularity] = 42 WHERE [Title] = ‘Generic Incident Request’
13.2 Managing Enumerations for Knowledge Base

Some enumerations are created and managed in the Cireson Portal and stored only in the ServiceManagement database. These enumerations are related to the HTML Knowledge Base but may expand to other types of enumerations in the future. The following steps describe how to manage those enumerations. All other enumerations are managed in the SCSM console and those changes are replicated to the ServiceManagement cache database via cache builder.

Navigate to the Enumeration Settings option in the upper right user menu.

On the main page you will see a list of editable enumerations. Click the one you want to edit.
Now the main edit screen displays. This page is comprised of three editable areas:

- **Enumeration list name.** Changes the name of the enumeration.
- **Enumeration list values.** Here you can add, remove, reorder and rename values. You can do this in the root level as well as any number of child levels deep.
- **Localized enumeration display strings.** Allows you to edit the localized string.

### Editing list values

**Adding Values** to current level in the tree: When the page first loads, the current level is the root level in the list. If you click the ‘Add Value’ button it will add a new item in the list at the root level. In order to add a sibling to a nested list you need to select an existing item in the level and click ‘Add Value’.

**Adding children:** Select the item in the list that you want to be the parent and click ‘Add Child’.

**Changing the name of values in the list:** Select the item you want to edit. With that item selected look at the ‘Localized Enumeration Strings’ area to see the display string for all available languages. In order to change it click on the text in the DisplayString column. To change the name in your local language make sure you select your languages LocaleId.
Deleting an item from the list: Select the item you want removed and click ‘Delete Value’.  
NOTE: deleting a parent with also delete all the descendant items.

Reordering items: Reordering is done my dragging the items to where you want them in the list.  
Mouse over the arrow icon to the left of the list items then click and drag it to where you want it.  
You should see a faint line appear as you are hovering over where you want to drop it. This line  
indicates where it will be placed. To move an item as a child of another item you can drag the  
item and drop it on top of that item. You should see a ‘+’ icon display when it will be put in as a  
child.

When done editing click ‘Save’ at the bottom of the page. After it saves you will see a green bar  
on the top indicating the changes were saved successfully

Disabling/Hiding Enumerations
Sometimes it is necessary to disable or hide an enumeration from the Cireson Portal. The two  
most common cases are:

1) Enumerations that are provided out of the box in SCSM that are in a sealed  
management pack and therefore cannot be modified but the enumerations are not  
wanted. Example: Service Request area enumerations or incident source enumerations.

2) An enumeration has been in use for some time and now needs to be “retired”.

An enumeration can be disabled by making a database update to the ServiceManagement  
database as follows:

- The first step is to determine the enumeration ID of the enumeration that you want to  
disable. You can refer to the How to Determine the Root Enumeration GUID of an  
Enumeration Data Type Property knowledge base article for how to find enumeration  
IDs in general. Although that article was written to help find the *root* enumeration ID,  
the techniques can also be used to determine the ID of any enumeration.

- Once you have the enumeration ID of the enumeration that you want to disable, run the  
following query in the ServiceManagement DB:

  UPDATE Enumeration SET Enabled = 0 WHERE EnumerationID = ‘the ID’

- Restart the Cireson Portal web site.
13.3 Additional resources

- Custom Form Example - Setting Support Group as ReadOnly - [https://vimeo.com/115581944](https://vimeo.com/115581944)
- Setting Up a SCSM Pre-Production/Staging Environment - [https://vimeo.com/127307593](https://vimeo.com/127307593)

For more advice and troubleshooting assistance, take advantage of the Cireson Community to find answers to your questions: