



Tungsten TotalAgility Architecture Guide

Version: 8.1.0

Date: 2024-10-01

TUNGSTEN
AUTOMATION

© 2024 Tungsten Automation. All rights reserved.

Tungsten and Tungsten Automation are trademarks of Tungsten Automation Corporation, registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Tungsten Automation.

Table of Contents

Preface.....	6
TotalAgility documentation.....	6
Training.....	6
Getting help with Tungsten Automation products.....	6
Overview.....	8
Web/App tier.....	9
Server side Windows Services.....	9
Database.....	9
Client-side components.....	10
Real Time Transformation Service.....	11
Core Worker Service.....	13
Background System (Worker) tasks.....	14
Execution through threads in thread pools.....	14
System Thread Pool.....	14
Job Scheduling System Task.....	15
Archive/Delete Items System Task.....	15
Create Exceptions System Task.....	16
Create Jobs System Task.....	16
Monitoring System Task.....	16
Perform Auto Activities System Task.....	16
Process Activities System Task.....	16
Process Session Timeouts System Task.....	17
Process State Actions System Task.....	17
KM and Xerox FtpFetch System Tasks.....	17
Ingest System Task.....	17
Capture Data Clean Up System Task.....	17
Process Capture Timeouts System Task.....	17
Device Management Cleanup System Task.....	17
Evaluate Jobs System Task.....	17
Retention Policy System Task.....	17
License Monitoring System Task.....	18
Upgrade Job System Task.....	18
Auto Activity Thread Pool.....	18
Lock activities.....	18
Process Activities Thread Pool.....	18

ProcessActivityTask.....	18
Upgrade Job Thread Pool.....	19
UpgradeJobsTask.....	19
Custom Job Thread Pool.....	19
ArchiveJobTask.....	19
CreateExceptionTask.....	19
CreateJobTask.....	19
EvaluateJobTask.....	19
DeleteItemTask.....	19
CaptureBatchTask.....	20
ProcessStateActionTask.....	20
IngestSystemTask.....	20
Configuration.....	20
Adding custom thread pools to CoreWorker.....	21
Performance Counter Usage.....	22
Transformation Server.....	23
Task Retriever.....	23
Task Dispatcher.....	23
Activity Processing Logic.....	24
Transformation Process Pool.....	24
Image Processing Process Pool.....	24
Ports and protocols.....	25
Custom forms configuration.....	26
Scaling and resiliency.....	27
TotalAgility Web and Application servers.....	27
Deployment options.....	27
Scaling.....	29
Clustering.....	30
Performance and sizing best practices.....	30
Transformation Server.....	30
Deployment options.....	30
Scaling.....	30
Performance and sizing best practices.....	30
License Server.....	31
Deployment options.....	31
Import Connector.....	31
Deployment options.....	31
Scaling.....	31

Performance and scaling best practices.....	32
Reporting service.....	32
Deployment options.....	32
Scaling.....	32
Performance and scaling best practices.....	32
Docker container deployment.....	33
Business continuity.....	35
Safe scripting in TotalAgility multi-tenancy.....	37
Overview.....	37
Allowed.....	37
Not allowed.....	38
Guidelines on coding for safe scripting.....	38
Configuration.....	38
Tenant boundary.....	39
Access registry.....	39
Terminate application.....	39
Stack overflow exceptions.....	39
Cookies in TotalAgility.....	41

Preface

This guide gives you an overview of the salient points of architecture in Tungsten TotalAgility, covering various deployments for an on-premise, on-premise multi-tenancy or Azure environment.

TotalAgility documentation

Access the full documentation set online, from the [Tungsten TotalAgility Documentation page](#).

You can also access the TotalAgility documentation in offline mode by downloading it from the [Tungsten Automation Fulfillment Site](#) for each language separately.

For a full documentation set, and how to access the documentation in offline mode, refer to the *Tungsten TotalAgility 8.1.0 Release Notes*.


Training

Tungsten Automation offers both on-demand and instructor-led training to help you make the most of your product. To learn more about training courses and schedules, visit the [Tungsten Automation Learning Cloud](#).

Getting help with Tungsten Automation products

The [Tungsten Automation Knowledge Portal](#) repository contains articles that are updated on a regular basis to keep you informed about Tungsten Automation products. We encourage you to use the Knowledge Portal to obtain answers to your product questions.

To access the Tungsten Automation Knowledge Portal, go to <https://knowledge.tungstenautomation.com/>.

 The Tungsten Automation Knowledge Portal is optimized for use with Google Chrome, Mozilla Firefox, or Microsoft Edge.

The Tungsten Automation Knowledge Portal provides:

- Powerful search capabilities to help you quickly locate the information you need.
Type your search terms or phrase into the **Search** box, and then click the search icon.
- Product information, configuration details and documentation, including release news.
To locate articles, go to the Knowledge Portal home page and select the applicable Solution Family for your product, or click the View All Products button.

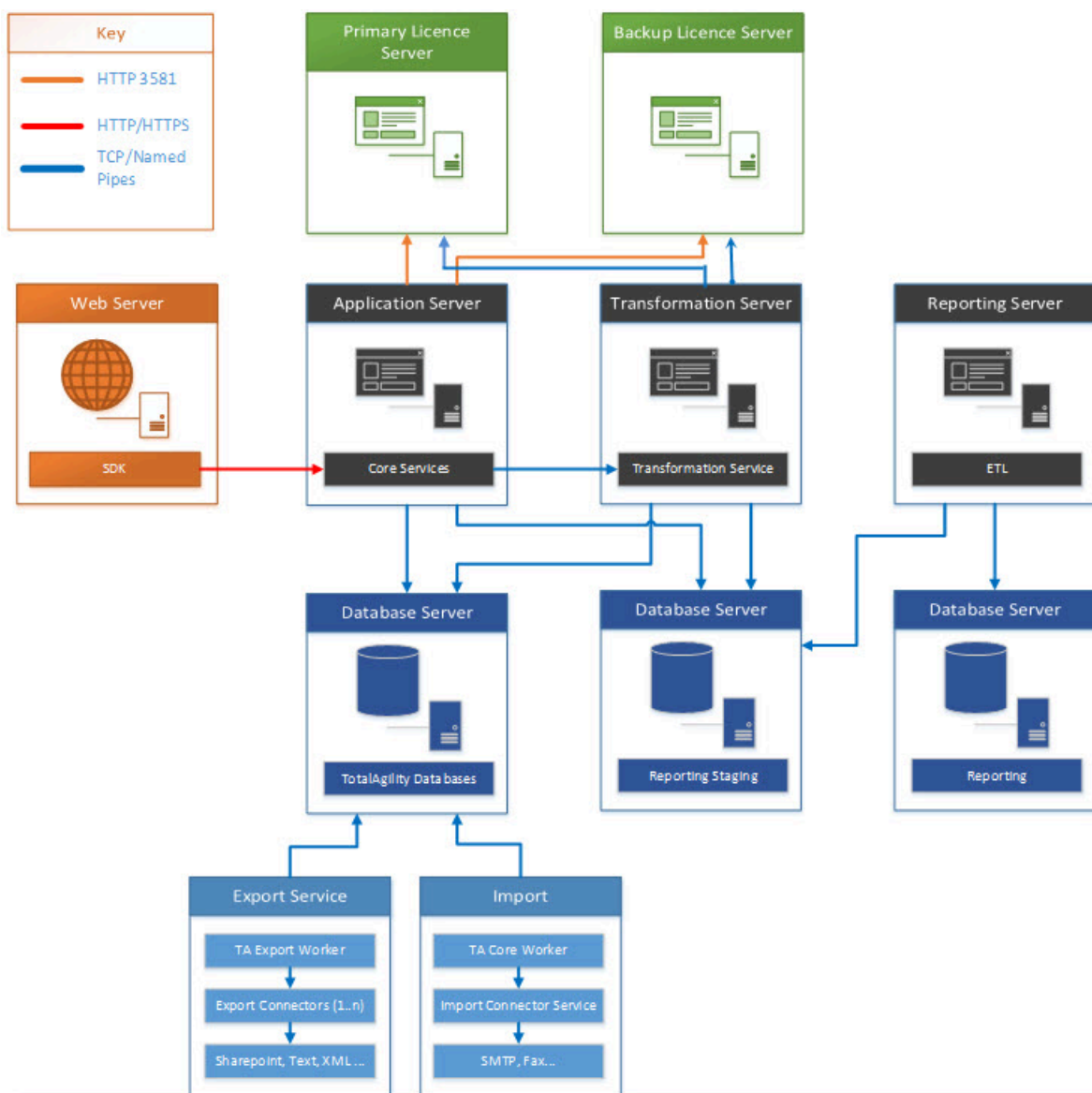
From the Knowledge Portal home page, you can:

- Access the Tungsten Automation Community (for all customers).
On the Resources menu, click the **Community** link.
- Access the Tungsten Automation Customer Portal (for eligible customers).
Go to the [Support Portal Information](#) page and click **Log in to the Customer Portal**.
- Access the Tungsten Automation Partner Portal (for eligible partners).
Go to the [Support Portal Information](#) page and click **Log in to the Partner Portal**.
- Access Tungsten Automation support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.
Go to the [Support Details](#) page and select the appropriate article.

Overview

The TotalAgility installation includes many components that communicate with each other as shown in the Logical Architecture diagram. For more information, see also the diagrams in [Scaling and resiliency](#).

Logical Architecture



Web/App tier

The web tier contains the SDK Services and provides a fully documented API set for users to call. There is no business logic contained within the SDK Services. All API calls to the SDK services are forwarded to the Core Services. The Core Services connect to the database and contain business logic. When a combined Web/App is deployed, the Core services reside on the Web server, and the calls from the SDK Services to the Core Services occur in memory. When a split Web/App is deployed, the Core Services reside on a separate Application server (the app tier).

Server side Windows Services

The server-side Windows Services include:

- **CoreWorker Service:** Responsible for executing the background system tasks, such as Perform Auto Activities, Evaluate Jobs, and more. See [Core Worker Service](#) for more details.
- **Transformation Service:** Polls the database for automatic capture activities to perform Image Processing, Separation, Classification, Extraction, and PDF Generation. As these operations are CPU intensive, this service must run on a dedicated server to perform its work when not running in Real Time Transformation Service mode. If running in Real Time Transformation Service mode, where it provides customers with enhanced features aimed at mobile/capture APIs, it works with in-memory documents to improve performance. See [Real Time Transformation Service](#) for more information.
- **Reporting Service:** Extracts the capture data from the Reporting Staging database, transforms it, and then loads it into the Reporting Main database. This service should be run on a dedicated server as it performs CPU-intensive translation of data.
- **License Service:** Decrements user and volume license counts.
- **Streaming Service:** Used to import and export packages in the TotalAgility Designer. You must use a Windows service as IIS only supports a maximum of 2 GB for a stream. This service runs on the web server. In a split web/app, it runs on both the web tier and the app tier.
- **(Import) Message Connector:** Enables documents to be imported into TotalAgility using the configured import connectors.
- **Export Service:** Exports documents to the customer's choice of store. Once exported, the documents can be removed from the main TotalAgility database.

Database

SQL Server is the supported database.

TotalAgility on-premise and on-premise multi-tenancy support SQL Server authentication or Windows authentication.

- **SQL Server authentication:** In this case, separate users are created for each tenant database and there are separate Live, Dev and Reporting schemas per tenant database. Use this type for higher security when deploying TotalAgility on-premise multi-tenancy.

- **Windows authentication:** In this case, each tenant database is accessed with the same Windows service account and there are separate Live, Dev and Reporting databases per tenant.

Client-side components

Tungsten Web Capture Service is responsible for image processing, bar code and patch code detection, file import from disk, and image normalization.

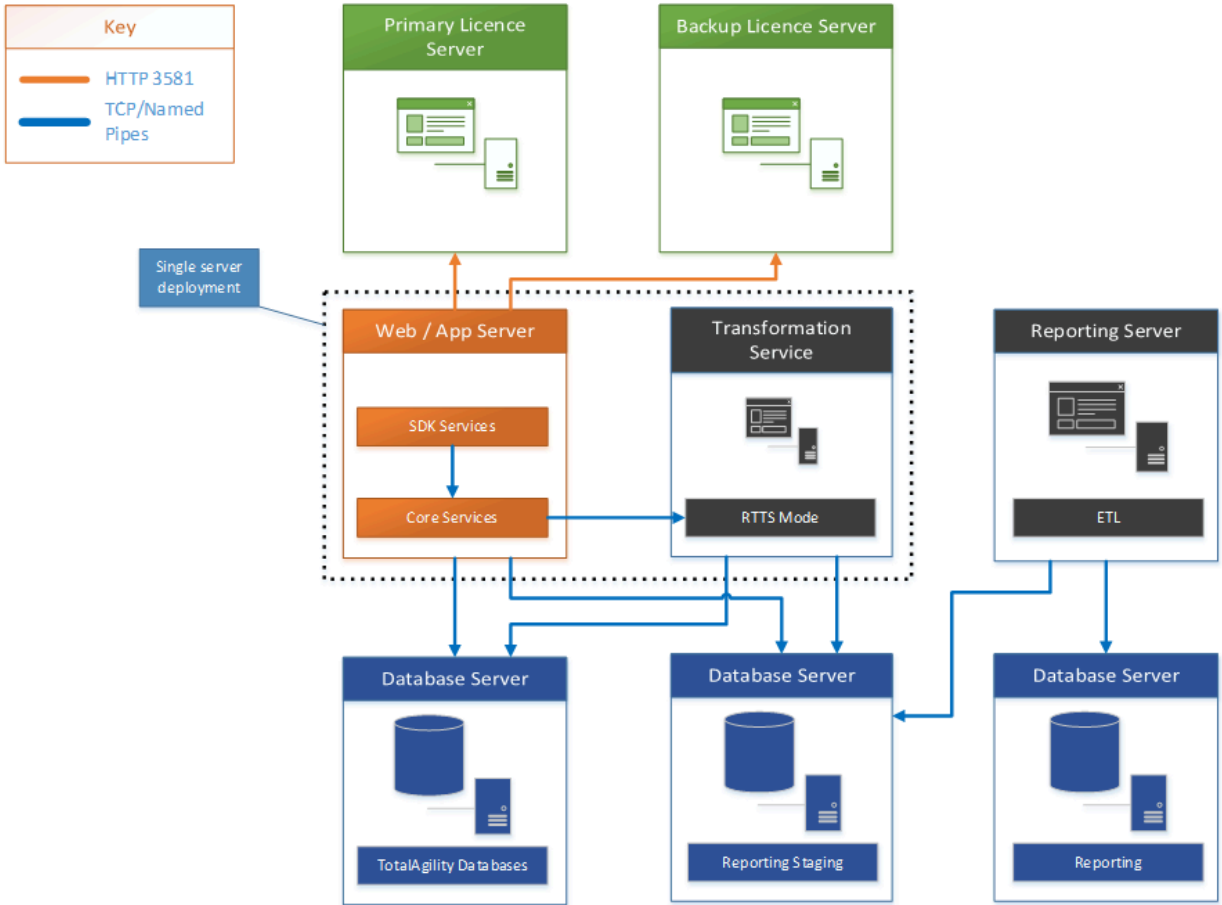
Real Time Transformation Service


TotalAgility can be installed in a specialized configuration mode known as the Real Time Transformation Service. In this mode, TotalAgility provides synchronous classification and extraction capabilities that reduce response time and improve performance by storing and processing document data in memory. You can persist this data in the document repository after classification and extraction.

TotalAgility supports real-time transformation and optimizes performance by:

- Supporting a configuration mode in Transformation Server that does not poll for pending capture activities and is used only for synchronous processing.
- Allowing Transformation Server to preload specific Transformation projects.
- Bypassing folder and document structure validation rules.

Real Time Transformation Service architecture

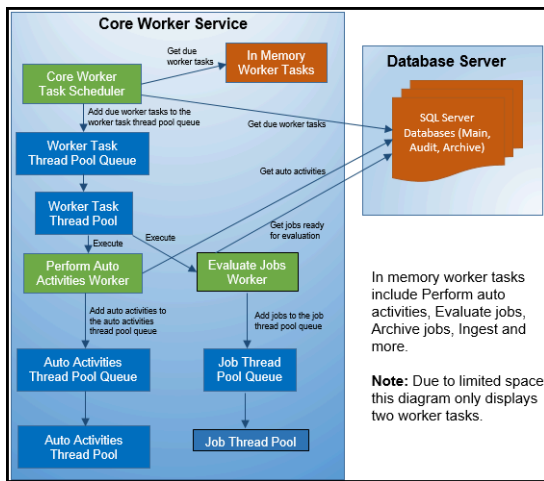


 Using the RTTS APIs is only supported when TotalAgility is installed in RTTS mode. The RTTS APIs will fail if they are not running in a TotalAgility RTTS installation.

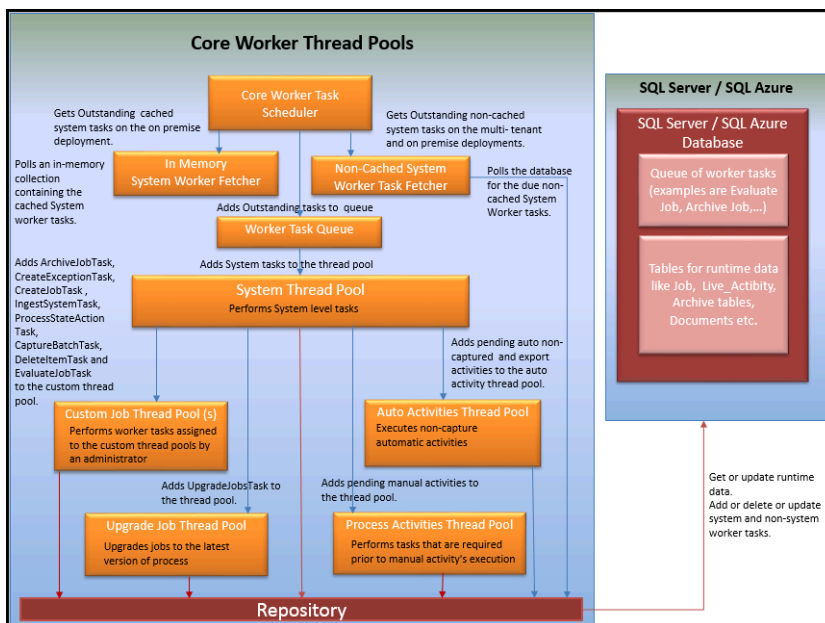
Core Worker Service

This section describes the inner workings of the Core Worker Service as shown in the diagram.

Core Worker Service



Core Worker Thread Pools



Background System (Worker) tasks

The Core Worker looks for non-capture System (Worker) tasks due for execution at preconfigured intervals. The frequency of their execution is based upon the interval, configured within the System Settings in the TotalAgility Designer.

In addition to a specific System task, some System tasks are also responsible for carrying out a non-system task. For example, the Evaluate Job System task is only responsible for evaluating jobs that are ready for evaluation, whereas the Archive Jobs System task not only archives completed jobs but also removes items that are marked for deletion by the Retention Policy System task.

Execution through threads in thread pools

TotalAgility system tasks are performed on multiple threads. These threads are managed through TotalAgility thread pools. Every thread pool is responsible for performing designated work. For each System task due, a corresponding entry is added to the process Worker Task Thread Pool Queue. When a thread in the process Worker Task thread pool becomes available, the item at the head of the queue is removed and allocated to the available thread. This thread then performs the (due) System Worker task. Once the System Worker task is completed, the thread becomes available again.

For example, the thread might perform the Auto Activities Worker task. This worker's task looks for automatic activities to perform. For each automatic activity found, an entry is added to the Auto Activities Thread Pool Queue. When a thread in the Auto Activities Thread Pool becomes available, the item at the head of the queue is removed and allocated to the available thread. This thread will perform the automatic activity.

Once the automatic activity is completed, the thread becomes available again. Essentially these automatic activities are executed asynchronously through threads in the Auto Activities Thread Pool.

Non-System tasks are also executed asynchronously through thread pools, in a similar fashion to the automatic activities described above. In this example, both the jobs to be archived and the retention policy items to be deleted are serviced by the Job thread pool.

System Thread Pool

The system thread pool performs the core system-level worker tasks. This section describes the different System (Worker) task types.

- In-memory / cached System Worker tasks

The Core Worker polls an in-memory collection containing the cached System worker tasks to identify which ones are due. When the Core Worker is scaled out horizontally across multiple servers, the in-memory Worker tasks can run concurrently. For example, the Auto Activities Worker task can run on both servers simultaneously.

The in-memory system tasks include:

- Perform Auto Activities

i TotalAgility Integration Server only executes the Perform Auto Activities and Ingest system tasks in-memory, it does not update due dates for these system tasks in the On-Premise Multi-Tenant or Azure server.

When the TotalAgility Integration Server executes the Perform Auto Activities system task for the assigned tenant on an On-Premise Multi-Tenant or Azure server, and if no activities are found, then it waits for the interval specified in the "NoAutoActivitiesFoundIntervalInSeconds" setting in the Core Worker configuration file before performing that system task again. By default, this configuration setting is 10 seconds. You can change this setting in Configuration utility, for more information, see *TotalAgility Configuration Utility Guide*.

- Evaluate Jobs
- Archive/Delete Items
- Ingest
- Create Exceptions
- Create Jobs
- Process Activities
- Process State Actions
- Non-Cached System Worker tasks

The Core Worker polls the database for the non-cached System Worker tasks to identify which ones are due. These types of System Worker tasks cannot execute concurrently across multiple servers. Example: Monitoring and Job scheduling.

Job Scheduling System Task

This system task does the following:

- Create jobs for an active schedule when configured due date is reached.
- Deletes schedules based on execution criteria. For example, when the configured end date is passed, or a specified number of jobs are created.
- Reactivates the jobs that are on hold.

Archive/Delete Items System Task

This system task does the following:

- Adds non-system worker tasks of the "ArchiveJobTask" type to the custom thread pool to archive jobs.
- Adds non-system worker tasks of the "DeleteItemTask" type to the custom thread pool to delete items such as jobs, documents, processes, and so on.
- Adds non-system worker tasks of the "TerminateJobTask" type to the custom thread pool to completely terminate the jobs.

Create Exceptions System Task

This system task adds non-system worker tasks of the "CreateExceptionTask" type to the custom thread pool to create exception jobs.

Create Jobs System Task

This system task adds non-system worker tasks of the "CreateJobTask" type to the custom thread pool to create custom jobs.

Monitoring System Task

This system task does the following:

- Monitors overrun activities and job cost and duration.
- Monitors overdue activities.
- Monitors underutilized resources.
- Monitors document storage limits in the Microsoft Azure environment.
- Raises the following exceptions when monitored entities fail on expected criteria.

Code	Name
EXP0001	Resource Inactive
EXP0013	Activity Due
EXP0006	Activity Overrun
EXP0004	Job Cost Overrun
EXP0005	Job Duration Overrun
Not defined	Storage Limit: The Monitor System Task checks for storage limits for the Database, and Capture Documents. This system setting is only available when running TotalAgility in an Azure environment.

- Updates the last monitoring run time.

Perform Auto Activities System Task

This system task does the following:

- Locks the specified number of non-capture automatic activities.
- Adds the auto activities to the thread pool for processing.
- Resets the timed-out non-capture activities.

Process Activities System Task

This system task adds non-system worker tasks of the "ProcessActivityTask" type for each manual activity to the process activity thread pool.

Process Session Timeouts System Task

This system task logs off the users when the session's last access time crosses the session timeout interval defined at the system level.

Process State Actions System Task

This system task adds non-system worker tasks of the "ProcessStateActionTask" type to the custom thread pool to execute state actions.

KM and Xerox FtpFetch System Tasks

This system task is responsible for fetching MFP files.

Ingest System Task

This system task adds system worker tasks of the "IngestSystemTask" type to the custom thread pool to import messages from connectors.

Capture Data Clean Up System Task

This system task adds itself to the custom thread pool to clean up outstanding capture data marked for removal, by deleting documents or folders.

Process Capture Timeouts System Task

This system task adds non-system worker tasks of the "CaptureBatchTask" type to the custom thread pool to check batch timeout for each capture batch.

Device Management Cleanup System Task

This system task performs KFS system clean-up on non-integration server deployments.

Evaluate Jobs System Task

This system task adds non-system worker tasks of the "EvaluateJobTask" type to the custom thread pool to evaluate the job.

Retention Policy System Task

This system task adds non-system worker tasks of the "DeleteItem" type to the custom thread pool to check the batch timeout for each capture batch.

License Monitoring System Task

This system task checks the current usage for each license type and raises the EXP0035 exception on over-usage.

Upgrade Job System Task

This system task adds non-system worker tasks of the "UpgradeJobsTask" type to the upgrade job thread pool.

Auto Activity Thread Pool

The auto activity thread pool executes non-capture automatic and export activities.

i The Export activities are executed by the Export service but not the Core Worker service.

Lock activities

By default, the CoreWorker service locks activities for processing in the order of activity due date and priority. However, you can change the default order by updating the CoreWorkerActivityOrder app key in the CoreWorker's config file.

If an order of job priority and then the job's expected finish time is selected, add the following index in the TotalAgility main database.

```
CREATE NONCLUSTERED INDEX [IDX_JOB_PRIORITY] ON [DBO].[JOB]
(
  [PRIORITY] ASC,
  [EXPECTED_FINISH_TIME] ASC)
INCLUDE ([JOB_ID])
```

This improves performance.

i The same .config file settings apply to the Transformation Server service also.

Process Activities Thread Pool

The process activities thread pool performs tasks that are required before the execution of manual activities.

ProcessActivityTask

This task does the following:

- Executes SignDoc activity
- Executes TCM (Tungsten Communications Manager) Compose and TCM Distribute activities

- Sends activity notifications
- Raises exceptions for invalid usable resources or when resources are not assigned to the activity.
- Performs manual work allocation based on the allocation algorithm selected at the server level.

Upgrade Job Thread Pool

The upgrade job thread pool upgrades to the latest version of the process.

UpgradeJobsTask

This task upgrades the job.

Custom Job Thread Pool

The administrator can add custom Job thread pools and set the number of threads in it and the worker tasks that it may perform. This helps in achieving better throughput and prioritizing the background tasks. Following are the non-system tasks performed by a custom job thread pool.

ArchiveJobTask

This task moves the completed and terminated job from the live database to the archive database when the server setting ArchiveFinishedJob is on.

CreateExceptionTask

This creates a job on the server and process level exception maps.

CreateJobTask

This task creates a job for the alert tasks.

EvaluateJobTask

This task does the following:

- Evaluates newly created job.
- Retry evaluation on evaluation failure during activity completion or job creation.

DeleteItemTask

This task deletes the following items as per retention policy.

- Job
- Form
- Process

- Skin
- Business Rule
- Classification Group
- Extraction Group
- Internal and External user
- Audit log
- Document
- TCM Pack

CaptureBatchTask

This task uploads timed out batches for capture processing.

ProcessStateActionTask

This task is responsible for executing state actions as described below.

- For state action of type job, creates jobs on the associated business process or case or case fragment maps.
- For state action of type event, executes associated business event. If an alert task is associated with the event, then it creates jobs on the alert task.

IngestSystemTask

This system task is responsible for importing documents from import message connector.

Configuration

You can optimize the system for specific workloads by configuring settings, such as the number of threads in a thread pool or the queue size for a thread pool. We recommend that you keep the default configuration settings; however, you can change them based on the customer's requirements. Changing a single configuration setting may not have the desired effect because many of the configuration settings are coupled.

For example, the following settings relate to the processing of automatic activities:

- **Thread Pool size for Automatic Activities:** The number of threads in a thread pool, that is, the maximum number of concurrent automatic activities.

For TotalAgility On-Premise, you can set the thread pool size for automatic activities in the **Automatic Activity** setting available at **System > System settings > System > Thread pools** in the TotalAgility Designer.

For TotalAgility On-Premise Multi-Tenant and TotalAgility in Azure environment, it is not possible to configure thread pools in the TotalAgility Designer, as they are not tenant-specific. For these environments, you can set the thread pool size for automatic activities in the **AutoActivitiesThreadPoolSize** setting available in **Agility.Server.Core.WorkerService.exe.config**.

- **CoreWorkerMaxAutoActivityThreadPoolSize:** The maximum number of automatic activities queued to be serviced by the thread pool.
- **System task interval for Perform Auto Activities:** The frequency at which the Core Worker polls the database for automatic activities.
- **MaximumNumberOfActivitiesToLock:** The number of automatic activities to look for when executing the Auto Activities Worker, and subsequently pass to the Auto Activities Thread Pool to perform.

Other settings include:

- **CoreWorkerIntervalInSeconds:** Dictates how frequently the Core Worker polls for due in-memory System tasks.
- **CoreWorkerDBIntervalInSeconds:** Dictates how frequently the Core Worker polls the database for System tasks that are due.
- **CoreWorkerWorkQueueSize:** Specifies how many System tasks are requested each time Core Worker polls the database.

Example

Consider increasing the **MaximumNumberOfActivitiesToLock** parameter from 16 (default) to 50. It may result in higher throughput of activities processed; however, if the Core Worker is not processing automatic activities quickly, the thread pool queue may simply reach its limit (default: 100). In this case, the number of activities entering the thread pool queue increases in size, but the number of activities being serviced by the thread pool remains the same. To remediate the issue, consider increasing the thread pool size and **CoreWorkerMaxAutoActivityThreadPoolSize** accordingly.

Adding custom thread pools to CoreWorker

You can specify new thread pools under "CoreWorker" in the "Agility.Server.Core.WorkerService.exe.config" file as follows:

```
<CoreWorker>
  <JobThreadPools>
    <add Id ="1" Size ="10" WorkerTasks ="ArchiveJobTask, CreateExceptionTask"/>
    <add Id ="2" Size ="10" WorkerTasks ="ImportSystemTask"/>
    <add Id ="3" Size ="10" WorkerTasks ="EvaluateJobsTask"/>
    <add Id ="4" Size ="10" WorkerTasks ="ProcessStateActionsTask"/>
  </JobThreadPools>
</CoreWorker>
```

In the above example, four thread pools are added, where each thread pool contains ten threads. The IDs must be unique. The list of worker tasks is passed as comma-separated values. Thus, the first thread pool can only archive jobs and raise exceptions. The second thread pool will be responsible for ingesting documents in TotalAgility. The third evaluates jobs and ingests documents through FTP. The fourth is responsible for processing state actions.

A worker task cannot be assigned to more than one thread pool. If assigned, the Core Worker detects and logs the error and prevents startup. The Core Worker also ensures that all the possible worker tasks are assigned to some thread pool. If a task is missed, the Core Worker logs an error and prevents startup.

By default, a clean installation of TotalAgility has a custom job thread pool set up to cater to all worker tasks, and the number of threads in the default pool is 4. On upgrade, the number of threads is set to the job thread pool size that was set in the Designer.

By default, a clean installation of TotalAgility has custom job thread pools set up to cater to all non-system worker tasks. If custom thread pools were set manually or through the configuration utility in a previous version, the same are available on upgrade. See "Add custom job thread pools" in *TotalAgility Installation Guide* for more information.

Performance Counter Usage

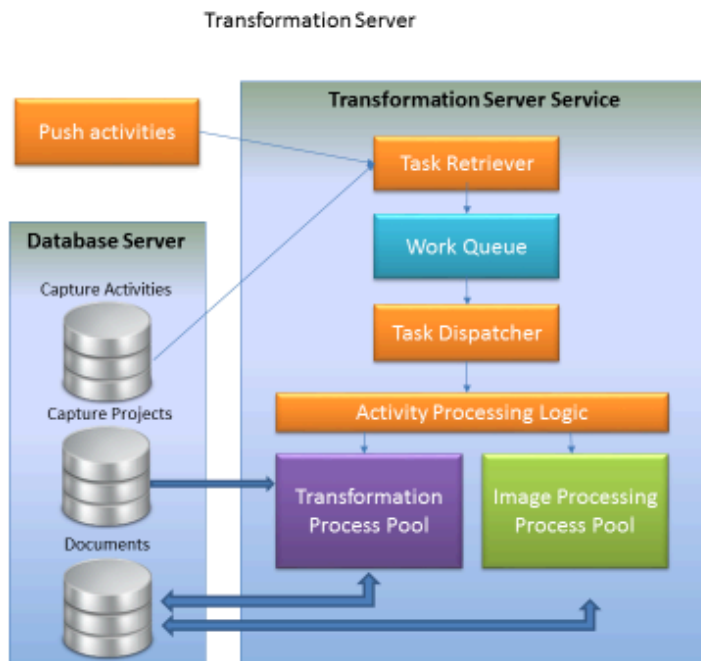
The Performance Counters are used to record application states and activities. Many are already included as part of the Windows operating system, and additional counters may be created programmatically. Counters are written and placed in an application where specific measurements need to be made, such as rate of throughput, count of items, and others. Once the counters are installed on a target system, an application can write values to them. These values can then be read in real-time using the Windows Performance Monitor, a Microsoft Management Console (MMC) snap-in. TotalAgility has multiple Performance Counters; the placement of these within the application is used to measure Key Performance Indicators (KPIs) of a number of critical services.

For more information on TotalAgility Performance Counters, see the following guides:

- *TotalAgility Administrator's Guide*
- *TotalAgility Installation Guide*
- *TotalAgility On-Premise Multi-Tenant Installation Guide*

Transformation Server

This section describes the inner workings of the Transformation Server as shown in the diagram.



Task Retriever

Fetches the capture work either from the database or directly across WCF calls from the TotalAgility Core Worker Service (known as Push activities). Work coming over the WCF channel is of high-priority and is placed at the head of the work queue to be processed when resources are available. Processing resources can be reserved for these high-priority activities to make sure there is no delay in processing them. The Task Retriever polls the database for more work when it anticipates that resources may be available for more processing.

Task Dispatcher

Maximizes the use of processing power on the server. The task dispatcher tries to consume all the server resources to maximize throughput unless it is configured to limit the number of documents that may be processed at the same time.

Activity Processing Logic

Handles the high-level logic involved in Image Processing, Separation, Classification, Extraction, Validation, and PDF Generation.

Transformation Process Pool

Includes processes that perform Separation, Classification, Extraction, and PDF Generation using the same core transformation modules as the Tungsten Automation Transformation product.

Image Processing Process Pool

Includes processes that handle image processing tasks that use eVRS, AtalaSoft, and other libraries to perform image conversion, image processing, scaling, bar code detection and separation, patch sheet detection and separation, and others.

Ports and protocols

Refer to the following table for ports and protocols used in the internal components. Also, refer to the Logical Architecture diagram in [Overview](#).

i If using Filestream in TotalAgility, see the Microsoft documentation for information on adding the default Filestream ports.

From	To	Protocol	Port
User	Web Tier	HTTP / HTTPS	80 / 443
Web Tier	App Tier	HTTP / HTTPS	80 / 443
Core Services in the App Tier / Core Worker Service / Transformation Server / Reporting Service / Analytics	Database	Named Pipes / TCP	1433
Core Services in the App Tier / Core Worker Service / TS / Reporting Service	License Server	HTTP / HTTPS	3581
License Proxy	License Server	HTTP / HTTPS	80 / 443
Core Services in the App Tier / Core Worker Service	Transformation Server	TCP	9001
App Tier	TRIM / SP / Dynamics CRM	HTTP / HTTPS	80 / 443
App Tier	MS Exchange	TCP	25

Custom forms configuration

You can seamlessly extend your forms-based solution using custom code where necessary using the following two options.


1. Use the **Custom Pages** and **Images** features in the TotalAgility Designer section of TotalAgility to upload custom HTML/JS and image files for use in any TotalAgility forms site.

This is the recommended approach as these pages and images are retained on upgrading TotalAgility.

2. Copy all custom pages to the following location:

```
<TotalAgility Installation>\TotalAgility\Forms\<sitename>
```

If you use a TotalAgility site, create a subfolder in this folder with the same name as the site and place all custom pages in this subfolder.

 With this approach, ensure that you back up the files before upgrading TotalAgility and then copy over again after upgrade.

We recommend that you create custom forms only when the functionality required is not achievable within the Form Designer, or when the form logic becomes too complex to maintain.

Scaling and resiliency

This section describes the deployment options and technologies available in TotalAgility to scale the installation to meet high-performance workloads and become resilient to interruptions and outages.

TotalAgility is a suite of stateless WCF.NET services. The architecture lets you scale your environment using standard load balancing technologies. These technologies include application clustering with WSFC (Windows Server 2022 and higher), Application Routing Request (ARR), Network Load Balancing (NLB), and other load balancing hardware and software.

TotalAgility allows each engine request to be serviced by any node in the farm that the load balancing software and hardware select. The state is maintained in a relational database, which you can scale further.

The web-based, mobile and MFP clients communicate with TotalAgility through the TotalAgility Web server using HTTP/HTTPS protocols.

We recommend that you install each server component onto a separate box except for TotalAgility Web server and TotalAgility Application server that may be installed together.

TotalAgility Web server, TotalAgility Application server, and Transformation Server can be scaled horizontally. You can improve the throughput of your installation by using multiple instances of these services.

TotalAgility Web server is an IIS-based web application that can easily scale using standard load balancing technologies. These technologies include WLBS, NLB, Application server, and other load balancing hardware and software. If you plan to install TotalAgility Web and Application servers separately, you also need to load balance traffic from the TotalAgility Web server to the TotalAgility Application server.

TotalAgility Web and Application servers

Deployment options

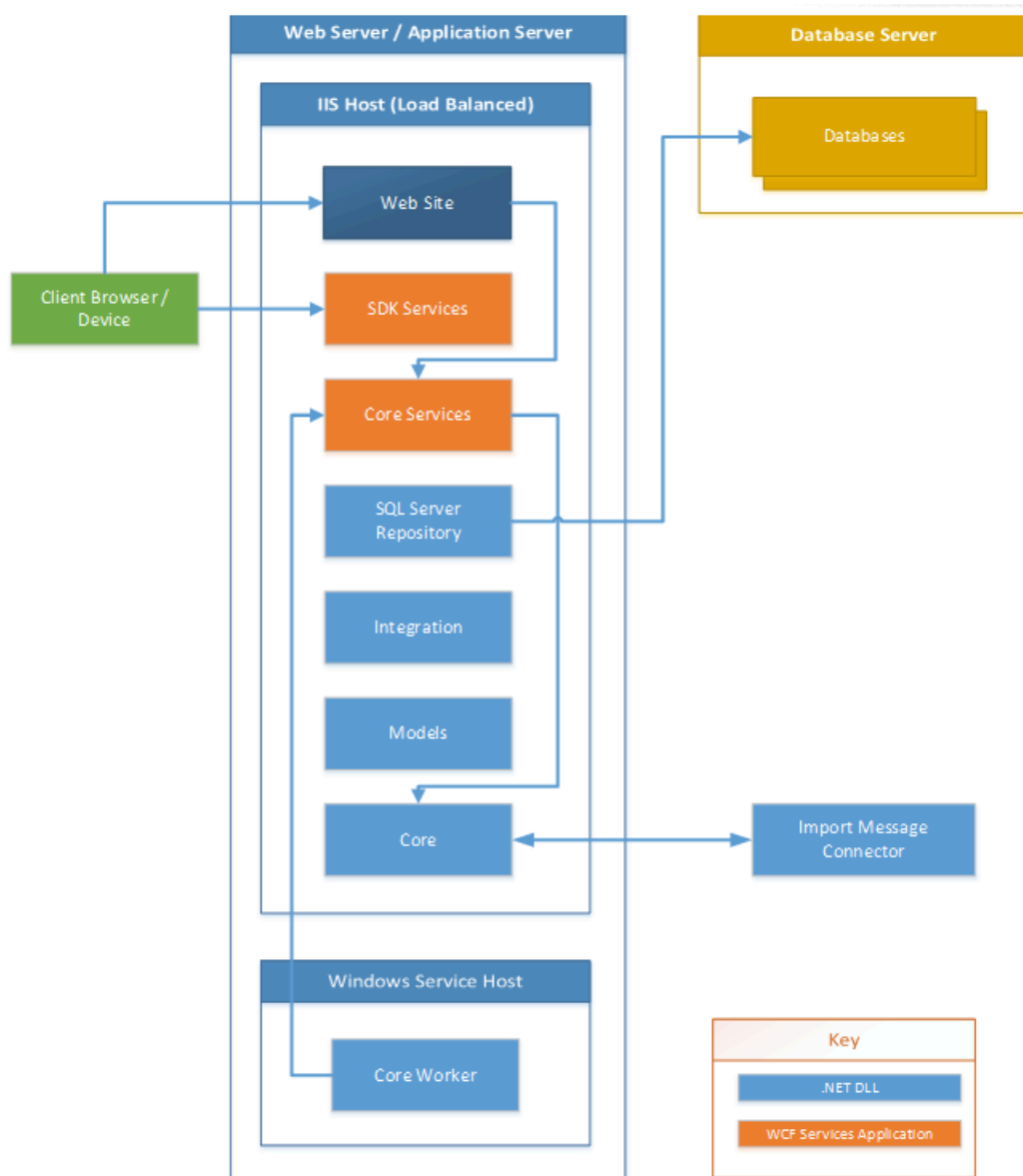
TotalAgility supports the following installation modes.

Combined Web and Application Server	Web and Application servers on a single computer.
Web Server only	TotalAgility Designer web pages, TotalAgility Workspace forms, and SDK services.
Application Server only	Core services

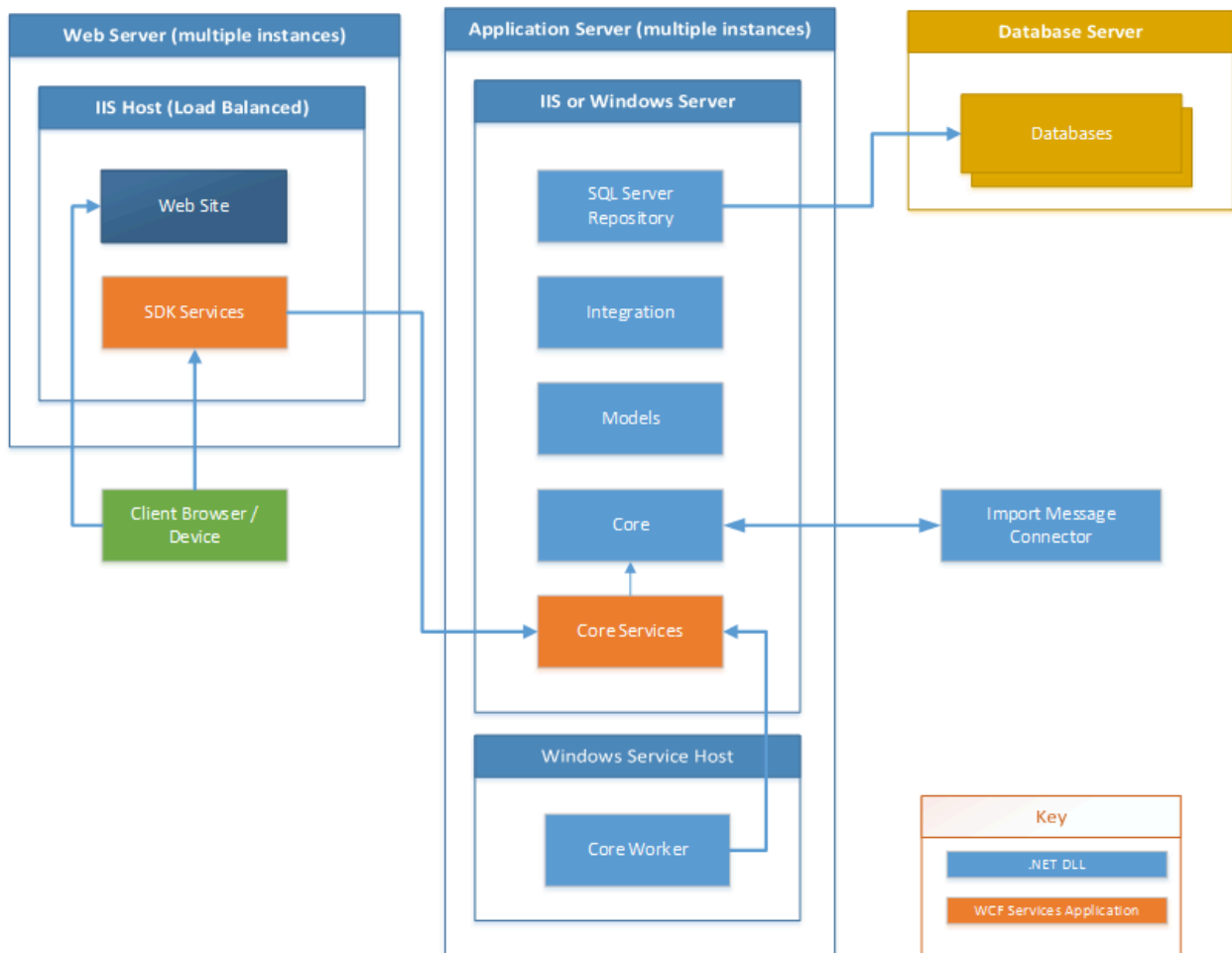
The TotalAgility Web and Application servers may be installed on the same computer or separately.

- Install the TotalAgility Web server and Application server on separate boxes if you do not need to access your enterprise internal network services (such as, databases and mail servers) from a Web server, or if you need to scale the TotalAgility Web server and Application Server separately.
- Although separating the TotalAgility Web and Application servers makes the installation more secure against Web attacks, having them installed on the same computer may result in better overall system performance. See the following images for the combined Web and Application server, and separate Web and Application servers.

Combined Web App Server



Separate Web App Server



Scaling

TotalAgility Web server is an IIS-based web application that can easily scale using the standard load-balancing technologies such as WLBS, NLB, Application server, and other load-balancing hardware and software. If you plan to install TotalAgility Web and Application servers separately, you also need to load balance traffic from the TotalAgility Web server to the TotalAgility Application server.

You can scale the TotalAgility Web and TotalAgility Application servers vertically (installing them on computers with more powerful hardware configurations, such as more RAM and more CPU cores) or horizontally (installing additional instances of TotalAgility Web and/or TotalAgility Application server service instances).

The TotalAgility Core Worker service on the Application server periodically polls the TotalAgility database for pending system tasks. Some examples of these system tasks are Perform Automatic Activities, Monitoring, and Job Archiving.

When the system does a high volume of background processing, we recommend that you scale out the TotalAgility Application servers. Each Application server will have a separate Core Worker service

instance. Each instance performs a share of the pending system tasks, thereby improving overall system throughput.

Clustering

Web servers can be hosted within a single or multiple NLB clusters, where all nodes are active at any one time, awaiting requests from the clients. NLB will route requests to the next available node (same IP address) within the respective cluster.

Use WSFC (Windows Server 2022 and higher) for the application server clustering where only one node is active at any one time. High availability is obtained through the creation of multiple WSFC clusters (each with its IP address). The application cluster is scalable in that extra nodes (up to sixteen) can be added for WSFC.

Performance and sizing best practices

Many factors may influence the number of TotalAgility Web/Application servers that you should install:

- Hardware configuration (number of CPUs, RAM, IO, and network throughput).
- Expected number of running jobs and complexity of TotalAgility processes.
- Expected load with background processing (unattended activities, system tasks, and more) that the TotalAgility Application Server should be running.
- Number of concurrent capture and non-capture user sessions (including mobile and MFP sessions).

We recommend monitoring the average CPU load for your TotalAgility Web and TotalAgility Application services. If the loads are higher than 70% for extended periods, you may need to deploy additional services. Otherwise, there may be performance degradation on peak loads.

Transformation Server

Deployment options

For better performance, the Transformation Server service should always be installed on a separate computer from other Tungsten TotalAgility services.

Scaling

To improve transformation throughput, you can install multiple instances of the Transformation Server service, each on a separate computer.

Performance and sizing best practices

Many factors influence the number of Transformation Server instances required to be installed:

- The hardware configuration of the computer where the Transformation Server is being installed (number of CPUs, RAM, IO, and network throughput).

- Number of extraction groups and document types defined.
- Number of documents planned to process hourly.
- Complexity of document types in terms of the number of pages, number of fields, recognition rules, and more.
- Using synchronous Transformation Server activity or fuzzy search queries.
- Using the Online Learning feature.
- Using a process for Trainable Document Separation.
- using layout or content classification (the latter requiring OCR, which consumes time).


If you plan to configure multiple Transformation Server pools, the throughput of every pool should be calculated to determine the number of servers in it.

License Server

Deployment options

TotalAgility uses a license server to manage all licensing requests. Your Tungsten Automation serial number and product code are locked to the computer.

TotalAgility supports one primary and one backup license server, and each license server needs its activation code and license configuration.

 The volume usage in the backup license servers can be very low (there only needs to be enough volume to last until the primary license server is restored).

If the primary license server fails, the system automatically switches to the backup license server.

The system can use the backup license server until it runs out of volume. When the primary license server is restored, TotalAgility automatically resumes using it.

Import Connector

Deployment options

The Import Message Connector is installed as part of the Application server. We recommend that you configure dedicated servers with the Import Message Connector when processing content in the production environments.

Scaling

As ingestion workload grows, additional Application servers can be installed, and their corresponding Import Message Connectors configured to accommodate the increase in workload.

Performance and scaling best practices

We recommend that you monitor the average CPU load for your Application server running the Import Message Connector. When CPU utilization is sustained at 70% for extended periods, you may need to deploy additional Application servers.

Reporting service

The Reporting service is installed by default when installing Application server services, but this can be changed during installation. We recommend that you install and run the Reporting service on its own dedicated server.

Deployment options

TotalAgility services require multiple Microsoft SQL Server databases:

- TotalAgility databases (such as Main and Archive)
- TotalAgility Document Repository database
- Tungsten TotalAgility Reporting Staging database
- Tungsten TotalAgility Reporting Main database

This list does not include databases installed by Tungsten Analytics for TotalAgility.

TotalAgility does not support databases where the default collation of the server instance is case-sensitive.

When the default collation is case sensitive, we recommend creating a new instance, ensuring the default collation of this instance is case insensitive. (Example: SQL_Latin1_General_CP1_CI_AS).

Scaling

TotalAgility supports database scaling technologies available to SQL Servers such as Always-On, Active-Active, or Active-Passive deployment of database instances, and scaling via increased hardware resources.

Performance and scaling best practices

You may require high storage volume for the Reporting and Document Repository. You can create these databases on separate SQL server installations, from each other, and from TotalAgility databases, to lower the load on the SQL server hosting the TotalAgility databases.

i For a TotalAgility on-premise deployment, the Reporting system task is only performed on one server at a time, so multiple Reporting servers can only be used for redundancy and not for horizontal scaling. For a TotalAgility on-premise multi-tenant deployment, each Reporting server can perform different Reporting system tasks for each tenant to achieve horizontal scalability.

Docker container deployment

You can deploy TotalAgility as Docker Windows containers instead of deploying directly onto Windows servers.


This avoids the need to use the TotalAgility installation program directly; instead, it is invoked automatically while building a TotalAgility container image. As all the prerequisites of TotalAgility are pre-installed on the container, the deployment is faster and easier.

A Docker host server can host multiple TotalAgility container types. These containers share the kernel with the host operating system and use fewer resources than a virtual machine.

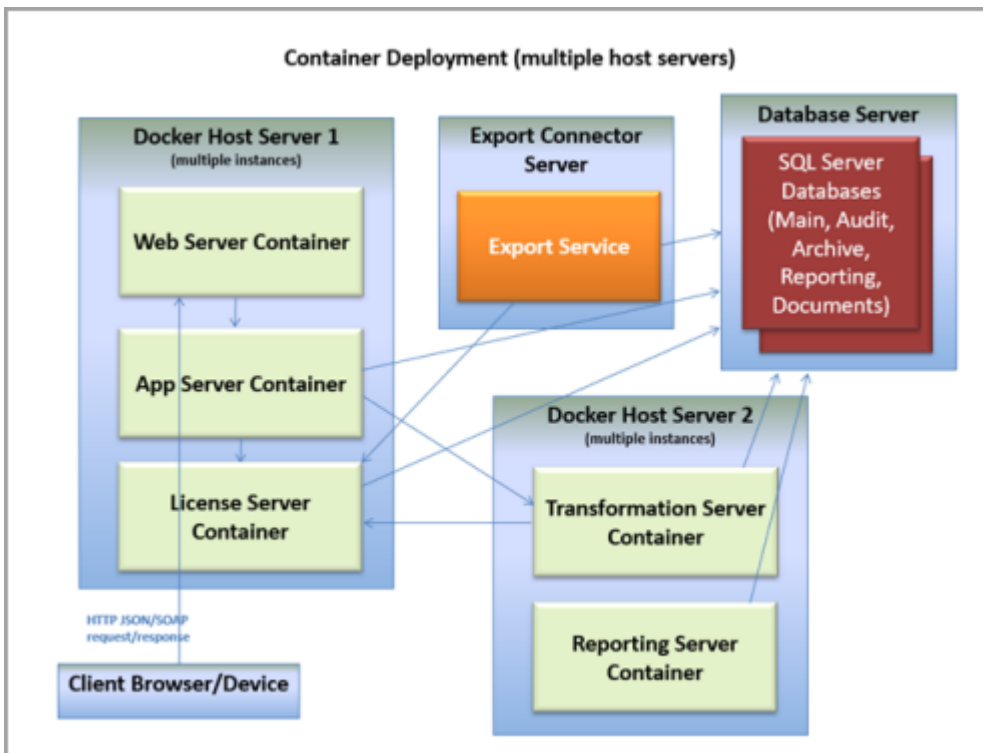
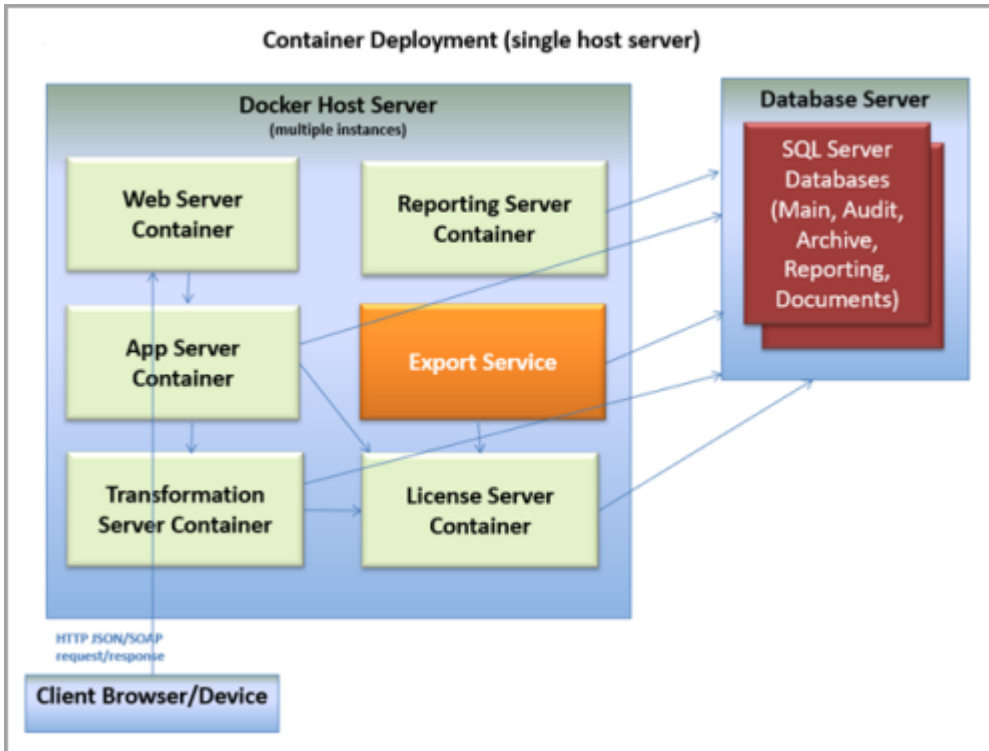
The container settings (including database connection strings) are passed at runtime through a Docker environment variables file.

You can use multiple Docker host servers to host a mix of different TotalAgility container types.

In addition, you can use a container orchestrator/clustering technology, such as Docker Compose, Docker Swarm, and Kubernetes, to manage TotalAgility containers.

 The Export service cannot run inside a container and must be run directly on a host server, because the export connectors have multiple dependencies which are not available in the container.

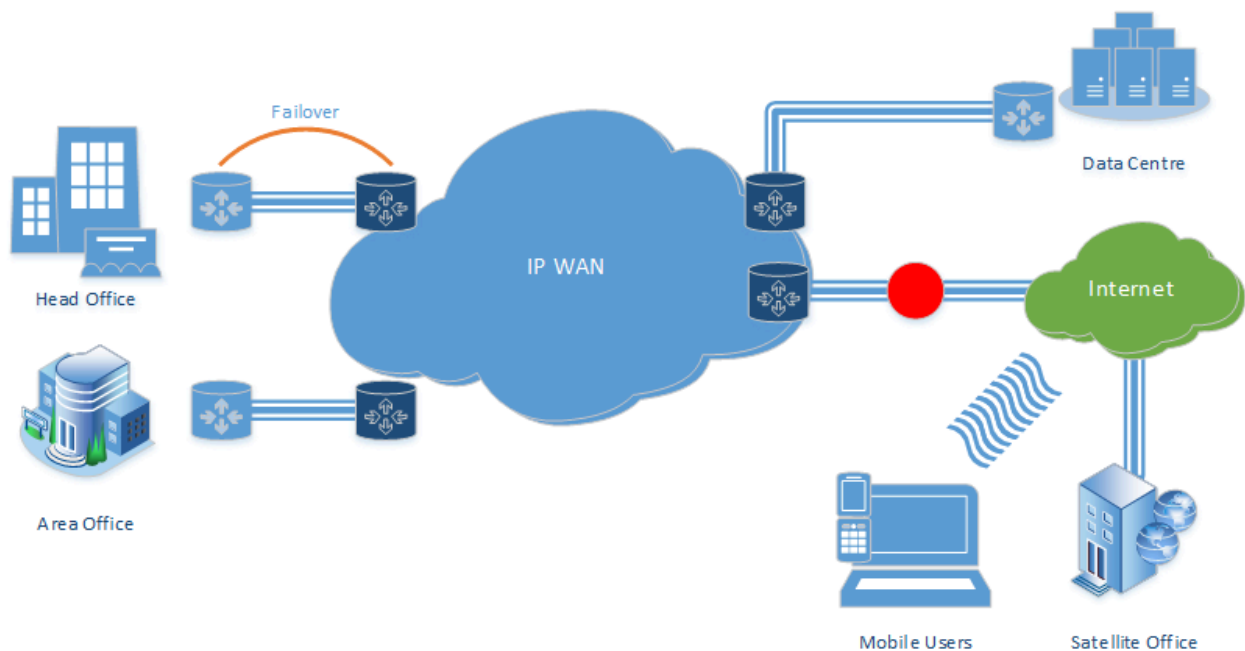
The following are some of the possible deployment configurations. For more information about building and running the TotalAgility Docker containers, refer to the *TotalAgility Installation Guide*.



Business continuity

In TotalAgility, business continuity means implementing a Disaster Recovery (DR) system as part of, or in lieu of, fault tolerance or High Availability strategy.

Usually, many DR options are temporary, and many DR infrastructures do not have built-in redundancy, powerful hardware, or fast network connections. In such scenarios, to resume the normal production level, we recommend that you develop or augment an existing DR plan after deployment of a new TotalAgility system, specifically if you have a vertically and/or horizontally scaled-out system. See the following sample WAN topology diagram.



i DR is concerned with total system failure and its subsequent reinstatement; it is not concerned with failover to a node.

Typically, most customers view DR as an overly expensive, relatively redundant system until they need it. You can consider one of the following approaches:

- Hot DR, which is fully automated.
- Warm DR with partial automation but still requiring a degree of manual intervention.
- Manual DR where every step requires human intervention.

You must perform the DR testing regularly and must factor it into your DR plan.

In TotalAgility, the following factors influence your DR plan:

- On-premise or hosted, or a combination of both

- Physical hardware or virtualization, or a hybrid of both
- Deployment topology
- Licensing costs
- Regulatory and/or business compliance
- Database deployment

Some businesses outsource disaster recovery to a managed service provider or cloud hosting company, as they cannot make the investment in location diversity for disaster preparedness, or do not have experienced IT staff (or no IT staff). Virtualization is often the de facto choice as shown in the preceding diagram.

As TotalAgility is a stateless-based application, the database is a key component. For an on-premise SQL Server, the following options are currently available.

- **AlwaysOn Failover Cluster Instances:** This option leverages the Windows Server Failover Clustering (WSFC) functionality to provide local high availability through redundancy at the server-instance level.
- **AlwaysOn Availability Groups:** This option is an enterprise-level, high-availability and disaster recovery solution. Use this feature to maximize availability for one or more user databases. To use this option, the SQL Server instances must reside on Windows Server Failover Clustering (WSFC) nodes.
 - The License Server database cannot be mirrored or load balanced.
 - Availability Groups do not support cross database transactions within the same SQL Server. No two databases in a cross database transaction may be hosted by the same SQL Server instance. So horizontally scaled dba (Live, Archive, Audit, documents, and others) have to reside on different groups.
- **Database Mirroring:** This option creates and maintains redundant copies of a database to ensure continuous data availability and minimize or avoid downtime. The future version of Microsoft SQL Server will not support database mirroring.

There is only one caveat for Disaster Recovery (Hot or Cold) with TotalAgility: The License Server cannot be load balanced or horizontally scaled. You must have a backup license server.

Safe scripting in TotalAgility multi-tenancy

Overview

TotalAgility has inherent support for multi-tenancy and ensures a tenant's data is isolated and cannot be accessed in any way by another tenant. TotalAgility can also safeguard itself from malicious code or unhandled exceptions in tenant's scripts or assemblies without affecting the execution of other tenant's assemblies. TotalAgility implements code access security (CAS) to protect tenants from fatal security issues.

Code Access Security (CAS) is a security technology developed by Microsoft to provide the ability to protect system resources when a .NET assembly is executed. System resources can be local files, files on a remote file system, registry keys, databases, printers, and others. Unfettered access to these types of resources can lead to potential security risks, as malicious code can perform damaging operations on them, such as removing critical files, modifying registry keys, or deleting data stored in databases.

TotalAgility supports the following script types:

- VB script: Allows a user to write legacy VBScript, based on COM technology
- C# Script: Allows a user to write C# code, the code is compiled into a .net assembly
- VB.NET script: Allows a user to write VB.NET code, compiled into .net assembly
- .NET assembly via .NET activity in a process map and form actions in Forms: Allows a user to call any method from any class in the assembly.

Allowed

TotalAgility allows you to do the following:

- Execute .NET managed code.
- Read and write to the current tenant folder and subfolders. However, other tenant folders are not accessible.
- Access the internet (call web services using the HTTP protocol).
- Access a public database if you have the credentials.
- Use security safe methods, that is, use .NET DLLs to call any methods from third party .NET DLLs that are allowed over security boundaries.



- You must use the ILMerge tool to merge all dependent DLLs to use them from a custom .NET DLL.
- You cannot refer to third-party DLLs from script activities.

- Use the .NET node and .NET form action to bypass security for TotalAgility SDK DLL. You cannot call TotalAgility SDK from script activities.
- TotalAgility only allows custom DLL with .NET Framework 4.0 and above. Any custom DLL with a .NET Framework version lower than 4.0 configured to a .NET activity or a .NET action fails the security test with the following error:

```
Assembly does not allow partially trusted callers.
```

Not allowed

TotalAgility does not allow the following:

- Use VB script execution for multi-tenant deployments, as CAS has security restrictions over the execution of non-CLR managed code.
- Access data from different tenants such as reading memory, file system, or database.
- Terminate or affect the process execution (from the current or other tenants.)
- Modify operating system systems, update the registry or reboot the machine.
- Call highly secured methods (called security critical methods) from any third-party libraries using the custom code.
- Call the TotalAgility SDK APIs using the custom code.
- Access the system folder or any folder other than the allowed folder.
- Let the tenant to read or write from registry.
- Terminate Coreworker exe/role or IIS.

Guidelines on coding for safe scripting

The following are some guidelines on coding for safe scripting.

Configuration

For TotalAgility on-premise multi-tenant deployments, custom code always executes in a secured mode. However, you can switch off security by changing the application setting, "SafeScripting" to false in the following configuration files:

- `Agility.Server.Web\Web.Config`
Execution of form actions, sync job, and business rule execution in test mode is affected.
- `CoreWorkerService\Agility.Server.Core.WorkerService.exe.config`
Execution of .NET script, business rule activity, .NET activity, and expression in an asynchronous job is affected.
- `Transformation.ServiceHost.exe.config` (Capture activity invoking business rule)
Turning off the Safe Scripting setting in the Transformation Server configuration does not impact the security mode for capture activities' execution. It implies that security is off for the business rules triggered during execution.

i It is not recommended to turn off Safe Scripting when tenants are able to create their own scripting in the TotalAgility Designer. It is not possible to turn off setting in TotalAgility Azure.

Tenant boundary

A tenant can access files only from its tenant folder and subfolders. This restriction helps in protecting critical system and tenant data. Code that tries to access the system folder or any folder other than allowed folder fails with the following security exception: Request for the permission of type `System.Security.Permissions.FileIOPermission`, failed.

- Malicious script for reading from system folder:

```
System.IO.File.ReadAllText("C:\Windows\System32\drivers\etc\host");
```

- Malicious script for writing to other tenant's folder:

```
System.IO.File.WriteAllText("Other tenant folder\ txt", lines)
```

Access registry

A tenant cannot read or write from the registry. Any code that tries to access the registry fails with the following security exception: Request for the permission of type `'System.Security.Permissions.RegistryPermission`, failed.

Malicious script for changing registry:

```
RegistryKey myKey = Registry.LocalMachine.OpenSubKey(registryKey, true);
myKey.SetValue(registrySubKey, subKeyDowordValue, RegistryValueKind.DWord);
```

Terminate application

Any malicious code that tries to terminate Coreworker exe/role or IIS fails with the following security exception: Attempt to access security critical method `'System.Environment.FailFast(System.String)'` failed.

Malicious script for terminating the caller:

```
Environment.FailFast(causeOfFailure);
```

Stack overflow exceptions

The Stack flow exceptions are unrecoverable memory conditions. These exceptions are not a security threat but can be potential risks to businesses. In such situations, you can only reduce the magnitude of impact as explained below:

- Coreworker role/exe or IIS will not terminate which ensures other tenants' work is not impacted.
- A tenant who is responsible for the exception is impacted.
 - If an exception is raised while executing the .NET action, then the other .NET actions need to be executed again.
 - If an exception is raised while executing the .NET activity, .NET scripts, or expression in a synchronous job, then the job needs to be executed again.

- If an exception is raised while executing the .NET activity, .NET scripts, or expression in an asynchronous job, then the activity will be executed again by the Core Worker service or role.
- All other work remains unaffected.

Malicious script for causing stack overflow exception:

```
new System.Threading.Thread(delegate()  
{  
throw new NotImplementedException("Dead");  
}).Start();
```

If you try to call Security Critical methods or the TotalAgility SDK APIs from the custom code, the code fails with the following security exception: Attempt to access security critical method.

Cookies in TotalAgility

Cookies are used in the following instances in TotalAgility:

- Session ID and relay state support of Federated security.
- Session ID used by TotalAgility Designer.
- Passcode for multi-factor authentication.
- Anti-CSRF token.
- Cordova version and other settings in MFP devices.

These session cookies are removed when the browser is closed, and they cannot be accessed using JavaScript (HTTPOnly).