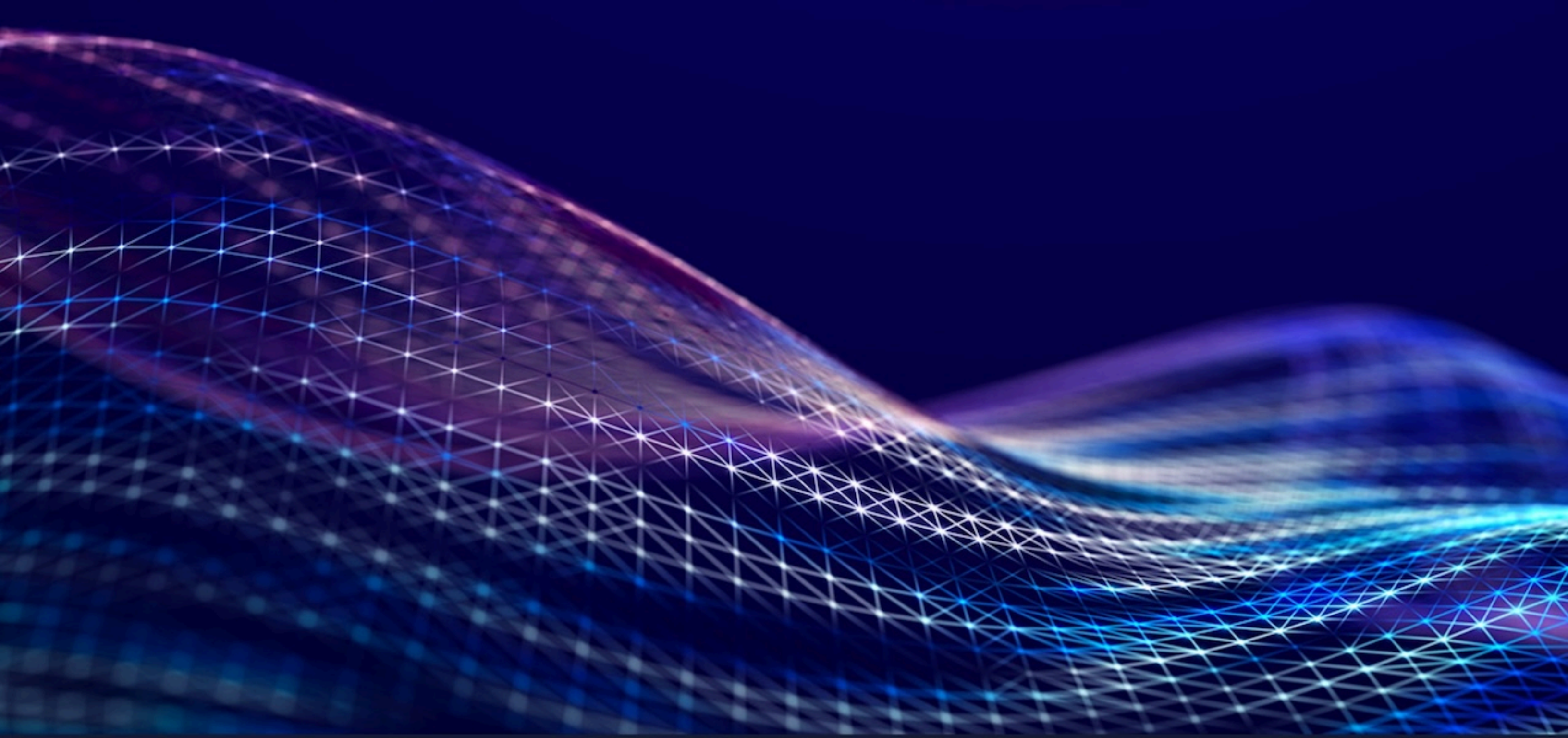


Tungsten TotalAgility

Best Practices Guide

Version: 8.1.0

Date: 2024-10-01

TUNGSTEN AUTOMATION

Table of Contents

Preface

This guide describes the recommended best practices that you must follow while using TotalAgility to improve performance, cost, maintenance, availability, and security.

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.

Chapter 1

Deployment topologies

This chapter describes the advantages and disadvantages of the most common deployment options.

The selected deployment option is completely driven by requirements, which typically include:

- Good performance
- Low cost
- Ease of maintenance
- Prevention of a single point of failure and support for high availability
- Adequate security

We recommend that you install the Transformation Designer and the Reporting Server on their dedicated servers. Both applications perform highly intensive CPU operations and being installed on their server ensures better performance.

Possible deployment topologies include:

- Combined Web App and TotalAgility Windows services
- Separate tiers for Web and App
- Combined Web App (where the services are disabled) with separate App Servers running the TotalAgility Windows services
- Separate tiers for Web, Core services App, and the TotalAgility Windows services App

The deployment topologies assume the Transformation Designer and the Reporting Server are installed on their servers.

In these topologies, the TotalAgility Windows services include:

- Core Worker service
- (Import) Message Connector
- Export Worker
- Streaming service

ⓘ Message Connector and Core Worker service must be installed on the same computer.

Combined Web Application and Tungsten TotalAgility Windows Services

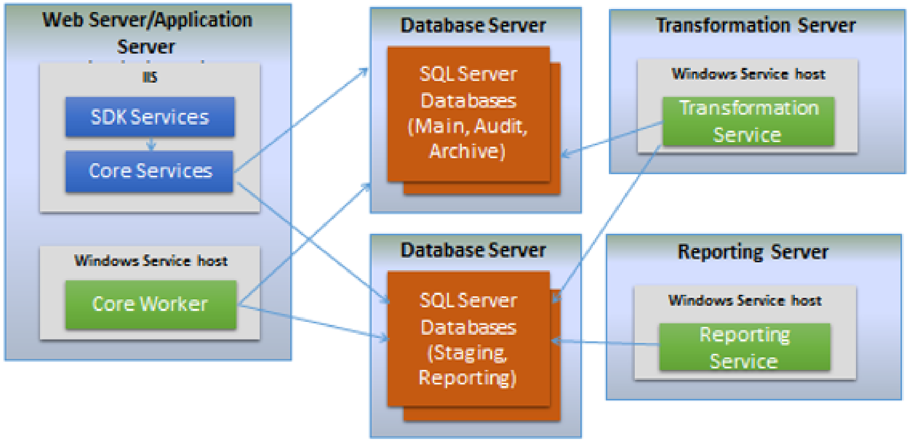
This is the most basic type of deployment where the exposed SDK services, the Core web services (called by the SDK), and the Tungsten TotalAgility Windows Services are all installed on the same server. This deployment is acceptable if the volume of non-capture automatic activities being executed is low.

Advantages:

- Calls to the SDK are most efficient as calls from the SDK to the Core services occur in memory.

Disadvantages:

- The Web server communicates directly with the database.
- The Web and Application server on the same server is not as secure as a split Web and Application where additional layers of security can be applied.
- The Web server is not dedicated to serving Web requests as it contains components that process background tasks.



Separate tiers for Web and App

In this deployment, the Web tier contains the exposed SDK Web services but does not contain the TotalAgility Windows services. The App tier contains the Core Web services and the TotalAgility Windows Services.

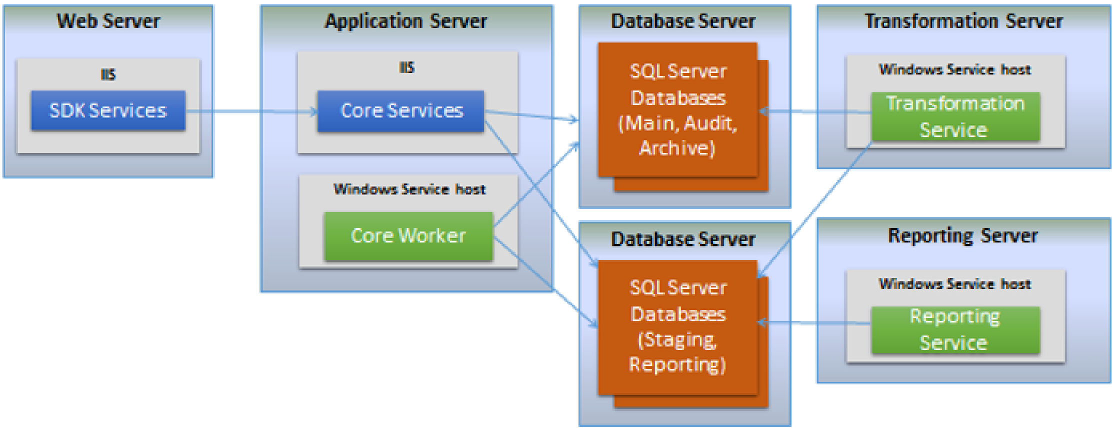
The Web tier does not connect directly to the database, providing an additional layer of security. The Core Web services communicate with the database. The additional security affects performance and the calls to the SDK APIs in this deployment are slower than in a combined Web Application deployment as a cross-machine call to the Core services is required.

Advantages:

- The Web server communicates with the Application server and not the database, which means additional layers of security can be applied to the App Server.

Disadvantages:

- The Core services reside on a separate server from the server where the SDK services reside and hence API calls to the SDK require cross-machine calls from the SDK to the Core services. Therefore, the calls to the SDK are not as efficient as in the Combined Web App and Tungsten TotalAgility Windows services deployment.



Combined Web App with separate App Servers running the Tungsten TotalAgility Windows Services

In this deployment, the Web Server contains the exposed SDK, the Core Web services, and the TotalAgility Windows services, as the TotalAgility installs them by default on a combined Web/App.

When compared to the Combined Web App, and Tungsten TotalAgility Windows services deployment, this deployment frees the Web server from the CPU load generated by the TotalAgility Windows Services, making SDK requests perform better. Similarly, the server containing the TotalAgility Windows Services is not loaded with SDK API requests, allowing it to also perform more efficiently.

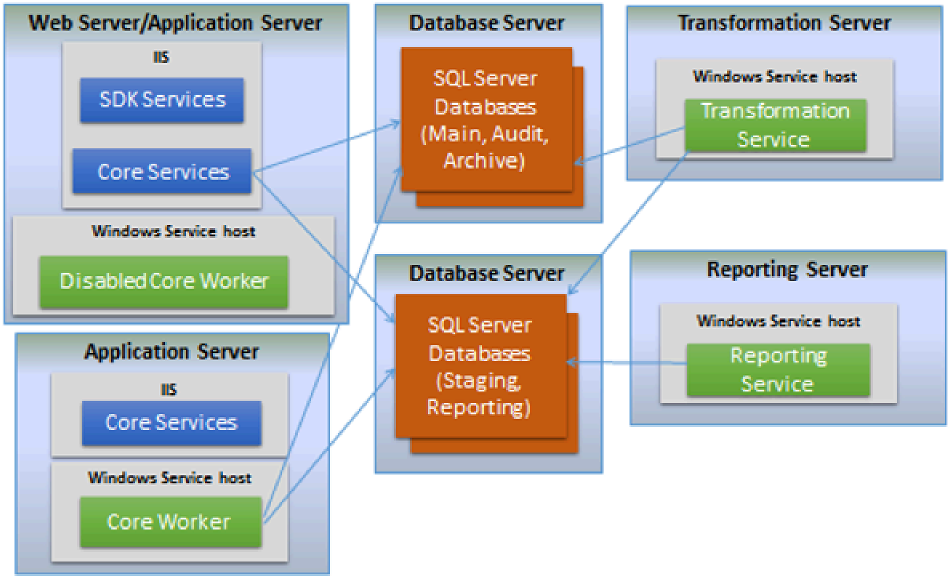
When compared to the Separate tiers for Web and App deployment, the SDK requests perform better in this deployment.

Advantages:

- Calls to SDK are most efficient as calls from the SDK to the Core services occur in memory.
- Dedicated App Server is responsible for processing background tasks.

Disadvantages:

- Web Server communicates directly with the database.



Separate tiers for Web, Core services App and TotalAgility Windows Services App

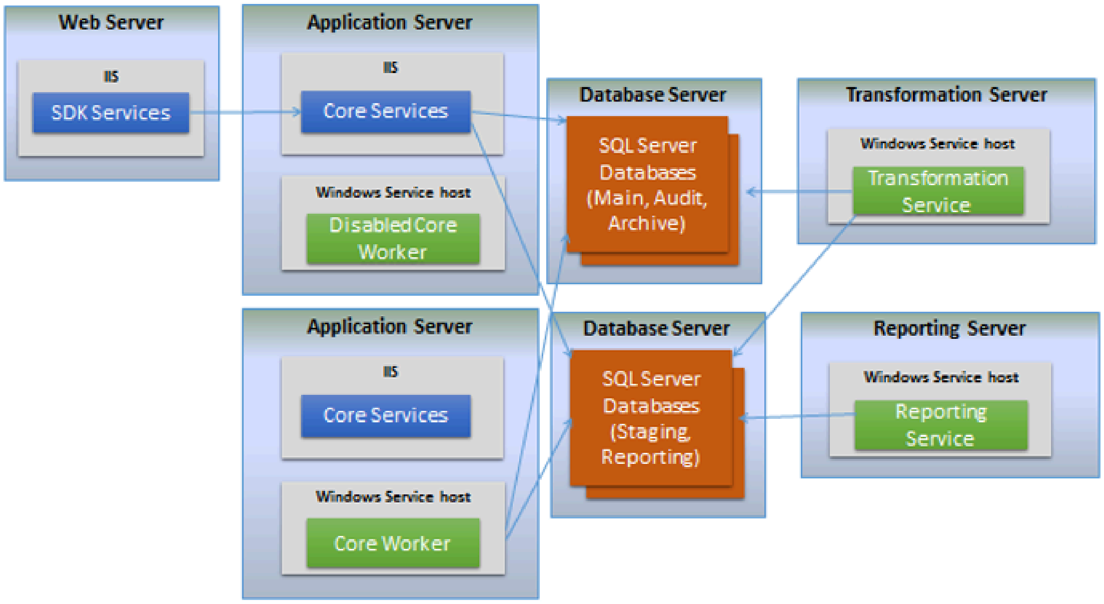
The main difference between this deployment and the Combined Web/App with separate App Servers running the TotalAgility Windows Services deployment is that Core Web services have their own dedicated server. In this deployment, the App tier CPU load is spread across a Web Server and an App Server; the Web server handles Web requests, and the App server handles background processing of system and non-system worker tasks.

Advantages:

- Web Server communicates with the Core services App Server, and not the database, which results in increased security.
- A dedicated App Server (containing the Core services) for receiving Web requests.
- A dedicated App Server (containing the TotalAgility Windows services) for processing background tasks.

Disadvantages:

- Calls to the SDK are not as efficient as in the Combined Web App and TotalAgility Windows services deployment, as the Core services reside on a separate server from the server where the SDK services reside. Therefore, API calls to the SDK require cross-machine calls from the SDK to the Core services.



High availability and scaling

When deploying any solution (not just TotalAgility), you need to be aware of the following:

- Single points of failure: The places in architecture where a single failure can cause the entire application to stop working.
- Scalability: Increased load.

Also, refer to the "Scaling and Resiliency" section in the *Tungsten TotalAgility Architecture Guide*.

High availability

To ensure high availability, we recommend the following:

- Install the TotalAgility Windows services across multiple servers to scale them horizontally. This approach improves performance as the load is distributed across multiple App Servers.

> It is not necessary for all the TotalAgility Windows services to be running on a single server. You can disable some and set them to run on a different application server (or servers for resiliency).

- Install the Reporting and Transformation servers on their dedicated servers.
- Load balance multiple Web servers in a high availability web farm configuration. This approach ensures that neither the Web tier nor the load balancer are single points of failure.
- Load balance the App Servers in deployments, where Web/App tiers are split, and the load balancer sits between the Web Servers and the App Servers.
- Introduce clustering, mirroring, or replication of the database to prevent a single point of failure.

- Use a backup license server to protect against failures.

Scaling

Improve performance by scaling horizontally and/or vertically to meet increasing load demands.

Vertical scaling: Hardware is modified to meet the demands of an increased load, such as more memory, or more cores on the server.

Horizontal scaling: The same software is run on multiple application servers, and you can have multiple application servers running the TotalAgility Windows services. Load balancing the Web servers can help improve the performance of your website.

See Possible deployment topologies for the advantages and disadvantages of different topologies.

Chapter 2

Performance

This chapter describes the recommended best practices for databases, forms, work processing, and other areas for improving performance.

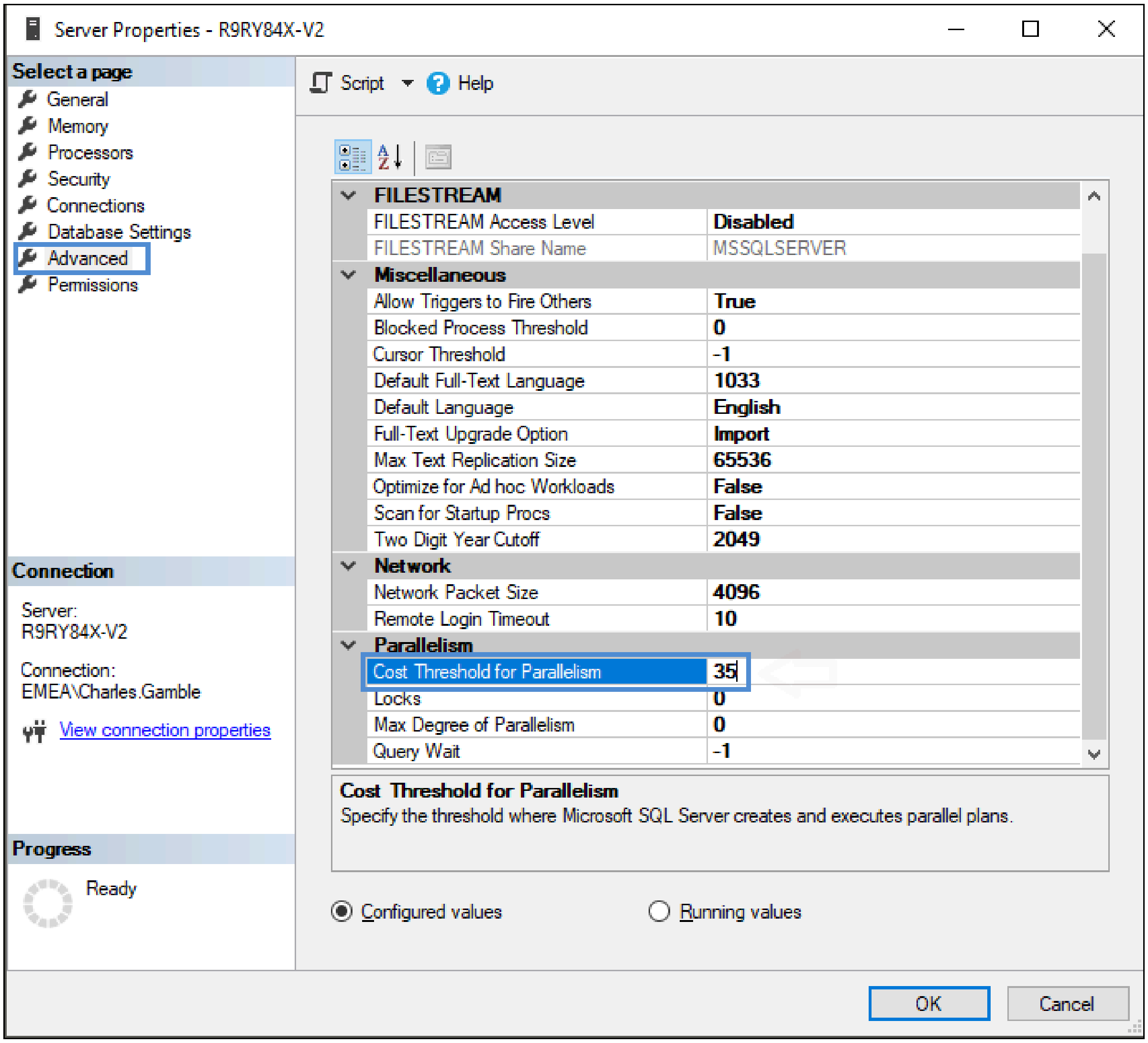
Database

This section describes the best practices for databases.

- Move finished jobs to the archive tables or purge them when the jobs are complete (or periodically). This approach ensures that the size of live database tables is minimized, so they do not grow continually.
- Switch the audit log off if you do not require it. This improves performance and reduces the database size.
- Define the retention polices at both the system and process levels so that the database growth is managed from the outset.
- All currently supported versions of SQL server offer *maintenance of indexes* online. While it is your decision to avail this facility, consider the following attributes as they affect the performance of TotalAgility:

    1. Server resource utilization increases (Memory, Disk IO) to complete the rebuild. The extra reads and writes are attributable to the second copy of the index that SQL Server creates during the initial phase of the index rebuild.
    2. Clustered indexes produce duplicate copies of both index and the data until the rebuild is complete, affecting performance on concurrent table activity. Non-clustered indexes marginally affect resource utilization.
    3. Transaction logs grow substantially, attributable to the higher frequency of REDO entries.
    4. Increased locking during the preparation and build phase ensures that other processes do not get an exclusive lock on the object while the index is being rebuilt.
    5. The last lock acquired, a schema modification lock (Sch-M lock), blocks all other concurrent access to the table (while the old index is dropped, and the metadata updated).

- When using SQL Server parallelism, in **Properties>Advanced** of the SQL Server, set the following property:
    - Under **Parallelism**, `Cost Threshold for Parallelism` = 35



Connection

Server:
R9RY84X-V2

Connection:
EMEA\Charles.Gamble

View connection properties

Progress

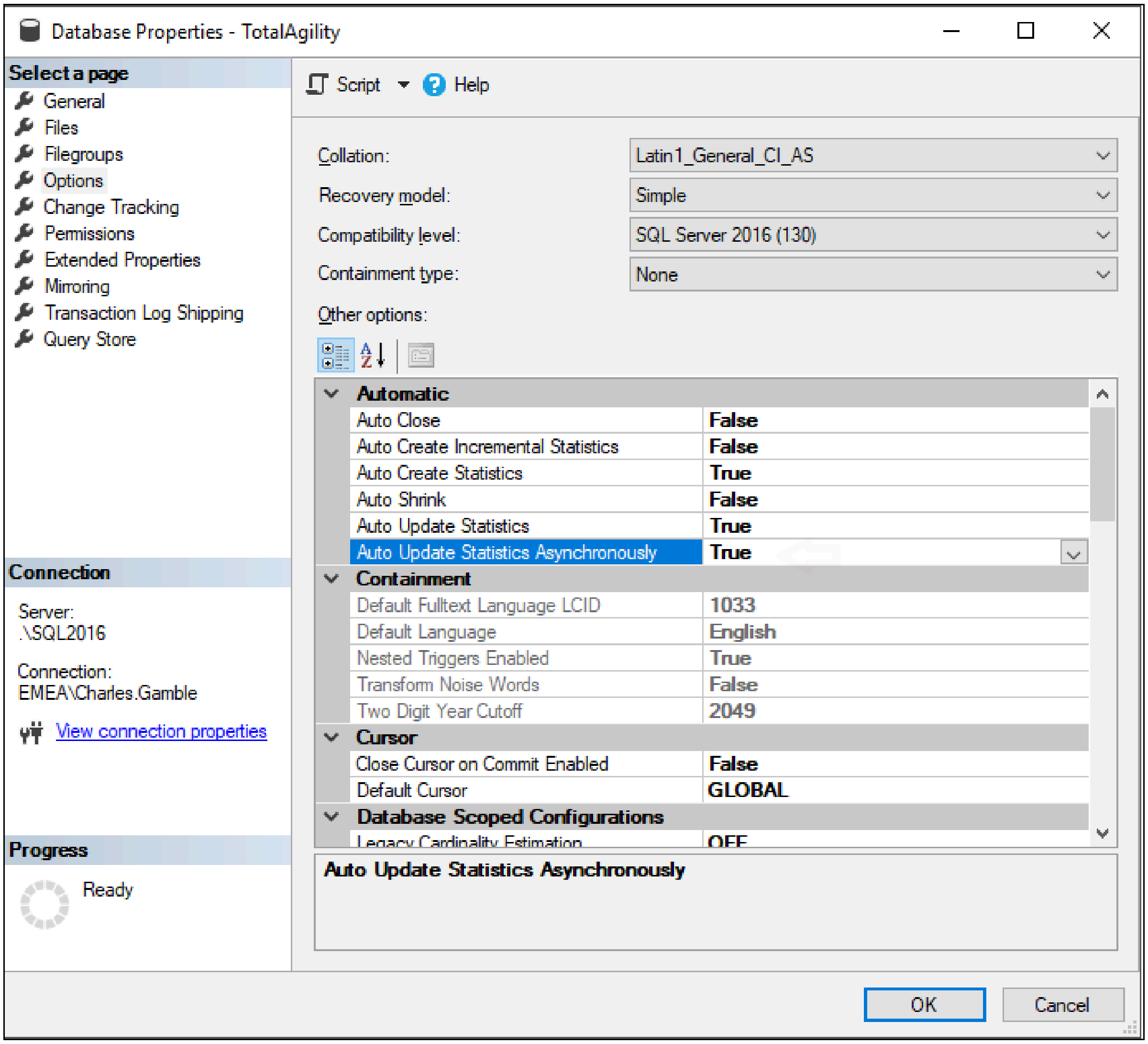
Ready

FILESTREAM	
FILESTREAM Access Level	Disabled
FILESTREAM Share Name	MSSQLSERVER
Miscellaneous	
Allow Triggers to Fire Others	True
Blocked Process Threshold	0
Cursor Threshold	-1
Default Full-Text Language	1033
Default Language	English
Full-Text Upgrade Option	Import
Max Text Replication Size	65536
Optimize for Ad hoc Workloads	False
Scan for Startup Procs	False
Two Digit Year Cutoff	2049
Network	
Network Packet Size	4096
Remote Login Timeout	10
Parallelism	
Cost Threshold for Parallelism	35
Locks	0
Max Degree of Parallelism	0
Query Wait	-1

Cost Threshold for Parallelism
Specify the threshold where Microsoft SQL Server creates and executes parallel plans.

● Configured values ○ Running values

- When using SQL Server 2016, in **Properties>Options** of each TotalAgility database, set the following properties:
  - Under **Automatic**, `Auto Update Statistics Asynchronously` = True.



Server:
.\SQL2016

Connection:
EMEA\Charles.Gamble

Collation: Latin1_General_CI_AS

Recovery model: Simple

Compatibility level: SQL Server 2016 (130)

Containment type: None

Other options:

Automatic	
Auto Close	False
Auto Create Incremental Statistics	False
Auto Create Statistics	True
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	True
Containment	
Default Fulltext Language LCID	1033
Default Language	English
Nested Triggers Enabled	True
Transform Noise Words	False
Two Digit Year Cutoff	2049
Cursor	
Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL
Database Scoped Configurations	
Legacy Cardinality Estimation	OFF

Auto Update Statistics Asynchronously

- Set the Maximum Degree of Parallelism (MAXDOP) of the SQL Server or each TotalAgility database based on the Microsoft recommendation for the number of cores/CPUs on the database server.

Form

Decide whether to use form cache, and if using what size. Consider the trade-off between the performance of regularly used forms and the memory on the client machines for storing these forms. We have set the default cache size appropriately.

Do not make large work queue calls or job searches. Any queries returning more than 50 rows impact performance.

Do not search with process variables. This practice is outdated and only available in APIs for backward compatibility.

Process

When all steps are automatic, use synchronous processes. The system converts these processes to .NET code that greatly reduces the load on the server.

When considering performance with a small volume, such as during demonstrations, be aware of the system behavior, which might cause performance issues. For example:

- If running an asynchronous map with 20 automatic activities, despite each activity taking milliseconds to run, there is at least 1 second of interval (default core worker polling interval) between them causing the map to take over 20 seconds.
- If performing an automatic capture activity, the Transformation Server goes to sleep for 30 seconds if there is inactivity, slowing down the map progress.

System monitoring

Use the Tungsten Monitor application monitoring software to get service level metrics, historical performance data and the real-time processing status of TotalAgility.

This software helps you better manage and assess your TotalAgility platform by providing real-time metrics on the operational health of your systems.

Capture Client

To achieve optimal UI responsiveness and performance while working with large jobs, we recommend you use the following:

1. **Chrome browser**: While TotalAgility tries to achieve the best possible performance for all supported browsers, TotalAgility achieves the best UI responsiveness with Chrome browser. This is especially true while working with large jobs or using undocked image viewer (or undocked web capture control).
2. **Remote scan**: To improve scan throughput at remote sites, deploy Scan Agent Service to enable asynchronous job uploads.

APIs supporting collection of object IDs or object indexes

In general, SDK API call does a database round trip. If you require multiple objects (like folder, document, or fields) access, we recommend you use the version of API that allows you to pass collection of object IDs or their indexes. Plural versions of APIs execute more efficiently than multiple calls of a singular version of the same API, as the database communication is significantly lesser.

Following are some examples of plural versions of APIs.

Document methods

- MergeDocuments
- RejectDocuments
- UnRejectDocuments
- UpdateDocumentsProcessingCompleted
- DeleteDocuments

Page and fields related methods

- RejectPages
- UpdatePages
- UpdateWords
- ValidateDocumentFields
- ValidateFolderFields

Chapter 3

Transformation Server instances

You can configure the Transformation Server instances when processing push activities. Push activities occur when a process designed to run multiple automatic steps returns to the user the next activity to do within the same job. In TotalAgility, the classification, extraction, image processing, and other such activities are purged to the Transformation Server instead of the normal processing where the Transformation Server polls for work.

You can process push activity on any Transformation Server instance that is allowed to process it.

- To restrict which Transformation Servers will process push activities, set "EnableSynchronousCalls" to false on servers that should not process push activities.
- For servers that should process only push activities, and not normal activities, set "SynchronousOnlyProcessing" to true.
- On servers that should process both normal activities and push activities, set "ReservedSlotsNumber" to 1 or 2 to prevent the occasional push activity from getting stuck in the queue.

Chapter 4

Security

Increase security using secure variables, assigning privileges, using virus scanners, applying Microsoft security best practices prior to installation, and other such measures.

Protect data at rest

Use the secure server variables to help protect sensitive information stored within the server variables.

Encrypt the database by using the SQL Server TDE to provide a further layer of security. Encrypting the database has an impact on performance; therefore, you must consider encrypting only those databases that contain sensitive information, such as the main TotalAgility database and the Documents database.

The TotalAgility configuration files contain references to connection strings, among other sensitive information. You can protect these details by encrypting the configuration files using the .NET utility. See the “Encrypt and decrypt the TotalAgility configuration files” section in the *Tungsten TotalAgility Installation Guide*.

Protect data in transit

Data in transit over HTTPS

Global variables used within TotalAgility forms are visible and editable in the browser. Use secure global variables in TotalAgility forms to protect sensitive data in transit. You can increase security by using HTTPS, specifically for deployment with public-facing websites.

You must have a valid certificate to use HTTPS.

When you require a public-facing website, deploy the web server in a demilitarized zone (DMZ) protected by a firewall. If traffic between the web server and the application server needs to be secure, we also recommend the use of HTTPS on the application tier. To provide additional security, place an additional firewall between the Web tier and application tier.

Data in transit with databases

While it is most important to protect data in transit over HTTPS, you can additionally secure data sent to and from the database by using SQL Server encryption for connections.

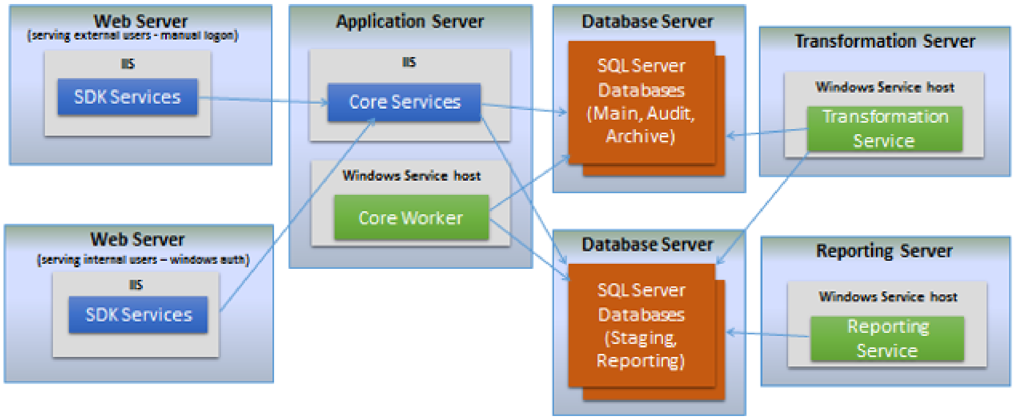
Privileges

Apply the "Principles of Least Privilege" to increase security.

Serve internal and external users

To serve internal users (within the LAN) and external users (internet or public-facing), different websites need to be configured on a Web server – one for the internal users and the other for the external users. For example, the website on the Intranet typically would not have the same security restrictions or requirements as the public-facing website.

Use different Web servers for different security requirements.



Protection against uploading malicious files in Scan Client

Since TotalAgility uses a Web architecture for handling file uploads through the Web browser, it is necessary to secure TotalAgility against the uploading of malicious files similar to securing a Web server. Based on the recommendations from OWASP (Open Web Application Security Project), we recommend the following best practices:

1. Ensure to implement the following Microsoft security best practices (see the Microsoft IIS 7 website) prior to installation:
    - Configure Web Server Security

- Configuring Security
- IIS Operations Guide
- Understanding Built-In User and Group Accounts
- IIS Security Checklist

2. Limit acceptable file sizes. Configure the file size limit through the Web server itself. Clearly understand the size of files you expect to receive before setting the limit.
3. Use a virus scanner on the server, deploy a firewall security appliance, or a Web application firewall that supports virus scanning. Example:
   `https://techlib.barracuda.com/waf/antivirusprotection`

Server hardening

Depending on your setup, you can apply some or all the following measures to protect against common server attacks.

Provide cross-site scripting protection: This header configures the built-in XSS protection that is found in most modern browsers. This header causes the browser to block a response if it detects an attack.

In IIS, add a HTTP Response header at the TotalAgility web-app level:

```
X-Xss-Protection=1; mode=block
```

Install URL Scan Module: This security tool restricts the types of HTTP requests that IIS processes. By blocking specific HTTP requests, the UrlScan security tool prevents potentially harmful requests from reaching applications on the server. It screens all incoming requests to the server by filtering the requests based on rules set by the administrator. Filtering requests helps secure the server by ensuring that only valid requests are processed. It can be configured to filter HTTP querystring values and other HTTP headers to mitigate SQL injection attacks while the root cause is being fixed in the application.

Reduce MIME Type Security Risks: To improve the security of your site against some types of drive-by-downloads, add the following header to your site at the TotalAgility web-app level in IIS:

```
X-Content-Type-Options=nosniff
```

Disable client cache at the IIS Server level: Open the Command Prompt as Administrator and run the following command:

```
appcmd set config /section:staticContent /clientCache.cacheControlMode:DisableCache
```

At the IIS level, add the cache-related HTTP response headers.

```
Cache-Control=no-cache; no-store; must-revalidate
Pragma=no-cache
Expires=0
```

Configure HTTP Strict Transport Security (HSTS): This mechanism helps to protect your website against protocol downgrading attacks and cookie hijacking. A protocol downgrade attack attempts to force a server to abandon an encrypted connection, such as https in favor of an insecure one (http). This may facilitate a cookie hijack and allow an attacker to access a valid session key. To implement HSTS, configure the web.config for your site as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
 <system.webServer>
  <rewrite>
   <rules>
    <rule name="HTTP to HTTPS redirect" stopProcessing="true">
     <match url="(.*)" />
     <conditions>
      <add input="{HTTPS}" pattern="off" ignoreCase="true" />
     </conditions>
     <action type="Redirect" url="https://{HTTP_HOST}/{R:1}" redirectType="Permanent" /
>
    </rule>
   </rules>
   <outboundRules>
    <rule name="Add Strict-Transport-Security when HTTPS" enabled="true">
     <match serverVariable="RESPONSE_Strict_Transport_Security" pattern=".*" />
     <conditions>
      <add input="{HTTPS}" pattern="on" ignoreCase="true" />
     </conditions>
     <action type="Rewrite" value="max-age=31536000" />
    </rule>
   </outboundRules>
  </rewrite>
 </system.webServer>
</configuration>
```

To use these configuration settings, make sure that IIS URL rewrite module is installed. Refer to the *Microsoft documentation* for more information.

Add content security policy header: This header allows you to define a whitelist of approved sources of content for your website. It restricts the assets that a browser can load for a site and acts as a countermeasure against cross-site scripting attacks. At the IIS level, set the following HTTP Response header:

```
Content-Security-Policy=default-src https: data: 'unsafe-inline' 'unsafe-eval'
```

Provide anti-clickjacking protection: You can control the behavior of iframes on your site to prevent clickjacking attacks. Set the X-Frame Options header to allow iframes from your site only:

```
X-Frame-Options=SAMEORIGIN
```

Disable OPTIONS verb: The OPTIONS verb can provide an attacker with information that is useful in facilitating further attacks. On the web server the built-in IIS module:

```
Request Filtering -> HTTP Verbs -> Deny Verbs…
```

Remove unnecessary headers: An attacker can use non-functional headers, such as X-Powered during a reconnaissance phase to identify a target platform. Remove such headers by using the IIS HTTP Response Headers module.

Protection against malicious database connections and web service URLs

If using the external database connections or web service URLs in a TotalAgility solution, it is necessary to secure TotalAgility against malicious database connections and web service URLs. In the Whitelist configuration of System settings in TotalAgility Designer, clear the "Allow All" option and specify each database connection string and web service URL that is required to be used by the TotalAgility solution.

Chapter 5

Database

This chapter describes best practices related to databases.

Sizing

Employ a database server fit for your requirements. The size of the database depends on your overall throughput of data, such as jobs and documents, and solution implementation. For example, solutions that include many long-running cases may require larger databases.

Ensure to employ the TotalAgility retention policies appropriately to manage the database growth.

If auditing is not required, switch it off so that database size does not grow unnecessarily.

Deployment

- Install the Reporting database on a separate database server from the core TotalAgility databases. This reduces the load on the core TotalAgility database server, ensuring optimal performance.
- The Documents database should reside on its own physical disk to limit contention of Disk I/O.
- Place the database and transaction logs on different drives, preferably separate physical drives.
- Distribute the MSSQL files over multiple logical drives. See the sample configuration.
  - SQL Server Installation – E:
  - SQL Server database data files – D:
  - SQL Server database index files – I:
  - SQL Server transaction logs – L:
  - SQL Server TempDB data files – T:
  - Backups – F:

  ⓘ Volume sizes and Input/Output Operations Per Second (IOPS) requirements for each of the items mentioned above vary, depending on the document and user volume on the system.

- Ensure all disks containing MSSQL files are formatted with 64K block size.
- Ensure all file growth for database files are set values and not the default percentages. Recommended: 256 MB for data, index, and transaction log files.

- Configure a service account for all SQL Server services. The MSSQL installer automatically assigns minimum privileges to the services account during installation.
- Install only the SQL Server Engine. You do not need additional components, such as Analysis, Reporting and Integration services.
- When installing MSSQL, make sure to apply all needed service packs and cumulative updates used by your organization.

  ⓘ Non-production instances of MSSQL should be on the same updates as Production.

- Ensure to enable TCP/IP Protocol.
- Configure TempDB:
  - By default, MSSQL places TempDB on the drive where MSSQL is installed. Move TempDB to its proper location.
  - Consider adding multiple data files to TempDB to avoid contention.
  - Manually grow TempDB to its final size (sizing varies depending on implementation).
  - Do not enable autogrowth for the TempDB files.
- Ensure mixed mode authentication is enabled.

Memory

Determine the maximum amount of memory you can assign to SQL Server by subtracting the memory required for the OS and any other instances of SQL Server (and usage on other system if the computer is not wholly dedicated to SQL Server) from the total physical memory. See the Microsoft website for more information on memory usage on SQL Server.

Maintenance

You can run an antivirus program on the MSSQL servers but exclude certain files to prevent performance bottlenecks caused by virus scans. See the Microsoft website for a full list of exclusions.

Schedule regular database maintenance plans to manage index fragmentation, statistics, backups, and archiving. Typically, the transaction log backups at 15-minute intervals are found to be appropriate, as these occur at a reasonable frequency (ensuring the log does not grow out of control) but not too frequently to impact performance.

The following Tungsten TotalAgility database tables become fragmented quickly depending on the system throughput, and need to monitor carefully to re-organize the indexes in short intervals and rebuild indexes in longer intervals. To avoid fragmentation, appropriate fill factors are used on these tables.

LIVE_ACTIVITY, LIVE_ACTIVITY_RESOURCE, EMBEDDED_PROCESS, WORKER_TASK, JOB, JOB_HISTORY, JOB_VARIABLE_HISTORY, LIVE_WORKQUEUE_DEFINITION.

For index issues on your current SQL DMV or STATS

If your current SQL DMV or STATS indicate an issue in index reports and sufficient SQL resources exist to handle optimizations, you must consider the following practices:

- If an index reports a high degree of fragmentation (consistently), you must set an initial fill factor of 80% (based on Microsoft best practice) for any index rebuild. Review it with gradual reductions to the fill factor to achieve the desired throughput (reduced page splits and reduced IO).
- If an index consumes one million plus pages, evaluate if enabling Large Page Allocations (8KB to 2MB) is feasible given the number of physical resources available (especially memory). This can reduce the number of physical pages and how they are loaded into memory thus having a great impact on performance.
- If an index with a sequential GUID / identifier (PK) has a high throughput, try rebuilding with `OPTIMIZE_FOR_SEQUENTIAL_KEY = ON`. This can control the rate at which new threads are allowed to request the latch, favoring sessions that are likely to keep the throughput high and also reducing memory pressure.

Resiliency

Consider the database resiliency by employing SQL Server clustering, replication, or mirroring. See the High availability section.

Chapter 6

Generative AI provider configuration

Use the following practices when configuring a Generative AI provider.

Models

- When configuring Azure OpenAI as a provider, the endpoint already contains the model name. Therefore, you can skip entering a model name.
- Model providers like Microsoft and OpenAI may deprecate models. Usually, the deprecation plan is announced well ahead of time. Make sure you plan your model testing and switch the model in time, so your production environment is not impacted. Refer to the Microsoft and OpenAI sites for the model retirement dates.

Generative AI provider for extraction

For the Generative AI provider configured for extraction, we recommend the following practices:

- Set the temperature to zero.
- Do not specify an imaging model, as the Copilot for extraction sends only the text to the LLM and ignores the imaging model.
- The Seed setting for OpenAI models is designed to give more repeatable results. It may be useful for demo purposes depending on the use cases. However, do not to use this setting for production purposes, as the same document is not processed multiple times.

Highlighting limitations

Highlighting is not always exact. Often the highlight location is larger than the extracted text. If the data is not printed but inferred, the data highlighting may be inconsistent. Sometimes inferred data cannot be highlighted at all. For such cases, a small rectangle on the top left of the document is used for highlighting. Highlighting may not be possible with a text-based LLM. Therefore, consider highlighting as guidance to identify where the data was extracted from.

Chapter 7

Solution building

This chapter describes best practices for building a solution in TotalAgility.

When building solutions, concentrate on the non-user interface aspects, such as documents, processes, and rules first to avoid rework on UX at a later stage. TotalAgility presents the data without forms in most cases so the flow and interaction can be played back early and tested without the need to build lots of UX. This can also facilitate earlier system testing.

System settings

TotalAgility is installed with several system defaults. You must review and if required, change them to suit your production environment.

Refer to "System settings" in the *Tungsten TotalAgility Designer Help* for description and configuration of these settings.

System session ID

For each installation of TotalAgility, a unique system session ID is generated. The system session ID allows the execution of any secure SDK call.

You can regenerate the session ID if there is a security breach (example: if someone has used the session ID to access the system without approval) or you can set this value (example: if moving from a development to production environment, and you want to use the same session ID for both the environments).

> ⓘ If you update the value, you need to update the Web.config files. Refer to the "System > System settings > Logon and authentication > User sessions" section in *TotalAgility Designer Help* for more information

Session and batch session timeout

The timeout settings include the session timeout and batch session timeout.

Session timeout

The session timeout defaults to 1 hour. After this period (relative to the user’s last active date) the system automatically invalidates the user session in TotalAgility.

If you want the TotalAgility sessions to time out, set the appropriate value or disable the "Process Session Timeouts" system task.

If you use the session timeout, the session has a limited lifetime and expires after a period. We recommend that you set the value appropriately for the purpose and nature of the application, to balance security and usability so that the user can comfortably complete operations without the session frequently expiring.

You can also allow or restrict users from having multiple sessions. See Allow multiple user logons.

Batch session timeout

This timeout period defaults to 30 minutes and is specifically used when performing capture-related tasks, such as Scanning, Validation, Verification, and Document Review.

If you want Capture batches to timeout, set the appropriate value or disable the "Process Capture Timeouts" system task.

We recommend that you set the session timeout for your typical users, considering how long they take to complete capture-related tasks and how intensive the capture work is. For example, consider the number of documents they need to scan or validate at a time.

- The batch can timeout independently from the session timeout and should have a lower interval than the session timeout.
- When using sticky sessions, the Batch session timeout interval does not apply; instead, the ASP.NET session timeout interval is used. Therefore, we recommend switching off the system task "Process Capture Timeouts" when using sticky sessions. See the *Tungsten TotalAgility Administrator's Guide* for more information on sticky sessions.

Password format

This property is a regular expression that can control both the length and complexity of user passwords.

The default password length is 10 characters.

We recommend that passwords be at least eight characters long and must represent a combination of character sets.

Password hashing algorithm

We strongly recommend the preconfigured Scrypt encryption algorithm.

SHA-1 is also available; however, recent advances in cryptanalysis have detected weaknesses in the SHA-1 algorithm. Scrypt is much stronger.

Disable logon without password

A person with knowledge of the TotalAgility SDK (and the deployed TotalAgility endpoint) can acquire a TotalAgility session on behalf of any valid user through the knowledge of the target user’s username only.

To secure all non-authenticating "session acquiring" SDK methods, enable the "Disable logon without password" setting.

When this setting is enabled:

- Any API that acquires a session ID cannot be called without a valid password.
- Any existing logon calls without a password fail.

Therefore, you must consider your solution context and determine if there is an impact and enable the setting accordingly.

Reset password notification process

When using the manual authentication with passwords, set the system process that handles the password reset requests.

Ensure that your users have valid email addresses, and an SMTP server is configured.

Allow multiple user logons

To allow users to log on to multiple TotalAgility sessions at the same time, select the Allow multiple user logon setting. Refer to "System > System settings > Logon and authentication > Password and logon" in *TotalAgility help* for more information

By default, TotalAgility allows only one active session per user. If a user logs on from another browser or location, the first session is terminated.

With the multiple session support enabled, each user can have multiple sessions that act independently. For example, each session can time out or log out without affecting the other.

However, each additional session consumes a concurrent user license. So, while in some scenarios it may make sense to allow multiple logons to increase usability, a rule of thumb is to not allow more functionality than what is required.

If your users are never going to connect to more than one simultaneous session, disallowing multiple logons reduces the risk of attack from an unauthorized user.

It may make sense to disallow multiple logons due to licensing, as closing the browser without specifically logging out leaves the session open and consumes a concurrent license until the session times out.

Account lockout policy

Within TotalAgility, you can configure how the system deals with unsuccessful logon attempts when manually logging in with incorrect passwords.

We recommend using the system defaults.

Maximum number of logon attempts: This setting determines the number of failed logon attempts after which a user account is locked out. The threshold set is a balance between operational efficiency and security and depends on your organization's risk level. To allow for user error and to prevent malicious attacks, keep the setting above 4 and below 10 (the default value is 5) as an acceptable starting point for your organization.

Account lockout duration: This setting determines the number of minutes (the default value is 30 minutes) an account remains locked out before automatically becoming unlocked.

You can configure this value to 0 so that the account is never unlocked automatically. Though it may seem like a good idea; however, doing so can increase the number of requests to your administrator to unlock accounts that are locked by mistake.

Use business calendar

The TotalAgility Workspace has a global business calendar that you can use to set working and non-working days and working hours across all resources. Each resource can also have a personal calendar that is derived from the global calendar.

By default, the business calendar is turned off. When the business calendar is turned on, any date calculations, such as job durations and activity due dates, are performed relative to the working hours.

If your application needs to take dates into consideration or you have service level agreements to meet, we recommend that you enable the calendar capability.

Allow duplicate email addresses

TotalAgility permits you to allow or disallow the use of the same email address by multiple resources.

Use of the same email address for multiple resources would be appropriate in development and UAT environments for testing purposes. Furthermore, in a production environment, it is possible for multiple resources to be using the same distribution email address rather than individual email addresses. Similarly, multiple groups may use the same distribution email address.

However, if you have external users logging in with their email addresses, we recommend that you disallow duplicates.

Write to audit log

Auditing is turned on by default. While audit logging can be useful for monitoring server activity and performance, the audit data can increase the TotalAgility database significantly.

You can clear the "Write to audit log" setting to reduce the database size and increase performance.

However, if you do wish to maintain the audit log, configure the associated retention policy to delete the old audit log entries.

Archive finished jobs

Move finished jobs to separate tables and maintain them in a separate database to manage the database size.

The "Archive finished jobs" setting is selected default. As a result, once a job is complete, the system task "Archive jobs" moves the job to the archive table.

This ensures that your live jobs tables are kept to a minimum and do not continually grow.

We recommend that you move finished jobs to the archive tables or purge them when the jobs are complete (or periodically).

> ℹ If you do not require the history of some jobs, clear the "Record history" setting in the process (Process properties > History, reporting and execution tab).

Reporting

If you wish to perform analytics on your process data and ensure that the data is picked up by Insight, select the "Include in analytics" setting, per process (Process properties > History, Reporting and execution).

The Capture data is stored in a Reporting database. To control how often the ETL agent handles the system task (extract/transform/load) and transforms data from staging to the warehouse, you can configure the interval using the Reporting warehouse ETL agent system setting.

You can restrict this transformation to nightly hours so that it does not interfere with day-to-day performance.

Form cache

Form caching applies to desktop forms only. By default, the form caching is turned on.

We recommend that you enable caching of forms if the users use the same forms repeatedly. This improves loading performance as forms are cached on the client-side browser for faster loading performance.

The number specified is the number of most recently used forms added to the cache. For example, 10 means that 10 forms will be cached.

The default form cache sizes are:

- Form: 20
- Document: 6
- Folder: 10

Under general conditions, the default settings provide the best balance between performance and client storage requirements.

A form is added to the client cache when displayed. If the cache is full for a form type, the oldest accessed form is removed from the cache to make space for the new form.

Every time a form is loaded from the cache, its last accessed time in the cache is updated to prevent it from being removed from the cache.

This means the most frequently displayed forms are in the cache.

Refresh durations

Custom pages and images can be uploaded to TotalAgility and then used in forms. The image or custom page displayed at runtime is downloaded from the database and cached.

If the image or custom page is updated in the TotalAgility Designer, the latest version is not displayed to users until the cache is refreshed.

To control the frequency of refresh, two system settings are available: "Image refresh duration" and "Custom page refresh duration". Both the settings default to 120 minutes.

We recommend that you set the interval considering the likelihood of images or pages being updated. In a production environment, you can set this interval to a larger value as updates are less likely.

Skill level

By default, the server-based skill level is selected (System>System settings>Process>SLA and work assignment). This means that every task within a job checks the skill level of a resource; in this instance, the resource has one value that encompasses all processes. The resource can only work on activities which they have the appropriate skills for.

If you use the process-based skills, for every process the resource is assigned a specific skill. This may require more maintenance and is more intense on the database.

If you do not use the skill level, the database access becomes less complex and thus can increase performance.

Exception handling

Consider if there are any system level exceptions that the solution would benefit from. For example, when a call to a Web service fails, in addition to suspending the job, would you want something else to occur?

You can configure exception handling at the system level to handle exceptions regardless of the source or configure it specific to a process.

By default, the system does not handle exceptions; therefore, when an error occurs while processing a job, the job is suspended, and notification are not sent.

We recommend that if you have a high-value use case where you need to closely monitor and avoid any downtime, or you have time-critical work to perform, you should configure the exceptions appropriately to the purpose and nature of the application.

You can configure the exception handling process to be very general, such as only notifying that the exception has occurred, or tailor it to use the initialization data that is passed to the map. This approach is useful if you do not want to send notifications for every suspension, or if you want to customize a remedial action for a certain type of exception.

Handle document conversion exceptions

When you import documents in TotalAgility and convert them, for some document formats, the conversion is not supported. In such cases, the documents get moved from Message Connector but are not available in TotalAgility. This may result in data loss.

You can configure TotalAgility to send the documents, for which conversion is failed, in an email as attachments. Additionally, you can also send the reason for conversion failure. If needed, you can import such documents manually in TotalAgility.

You can handle documents conversion exception as described below:

1. Enable the **Reject documents on exceptions** System setting.
2. Add a decision node after the Document conversion activity to check if the value of FolderVariable.HasRejections/FolderVariable.Rejected is true.
3. If the outcome of the decision is true, add an Email node to send an email with rejected documents as an attachment which part are part of the FolderVariable.

For more information, refer to "Handle document conversion exceptions" in the *TotalAgility Designer Help*.

Retention policies

Good design and maintenance should advocate that the older and unused items are manually removed from the system to prevent unnecessary database growth.

Retention policies can automatically assist in minimizing the database size. By default, the system does not provide any retention policies, and therefore all artifacts are retained indefinitely in the database unless they are manually removed.

We recommend that you define retention periods to clean up unused items, manage database growth, and optimize performance.

Configuration management for team-based deployment

Developing solutions in a team-based environment requires the use of configuration management. In the absence of any direct integration between TotalAgility and any configuration management tools, the following approaches are available:

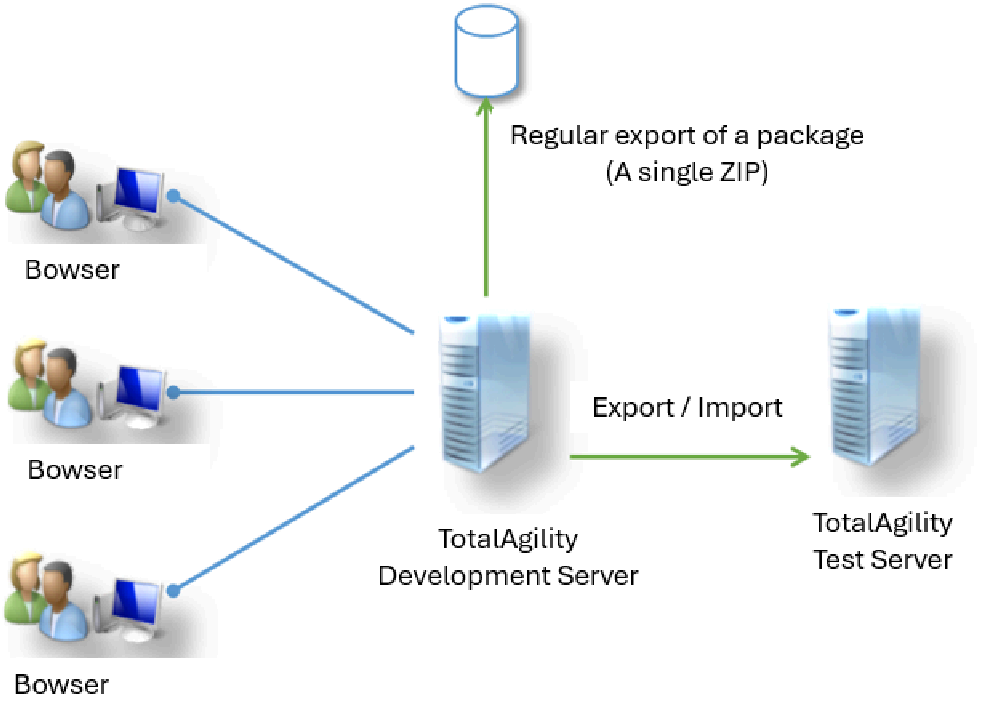
- Shared development environment: TotalAgility installed once on a central server
- Standalone development environments: TotalAgility installed multiple times locally

We recommend that you use the standalone approach because this approach is found to be the most productive with the least downtime due to check-in conflicts.

To determine which approach is best suited to your organization, review the analysis provided in the following sections.

Shared deployment environment

The development team browses to the TotalAgility Designer on the development server, making use of the out-of-the-box locking features to control access to forms, processes, business rules, and other relevant artifacts. This approach requires considerable discipline from the development team to minimize system downtime caused by conflicting changes that prevent other developers from continuing.



Advantages

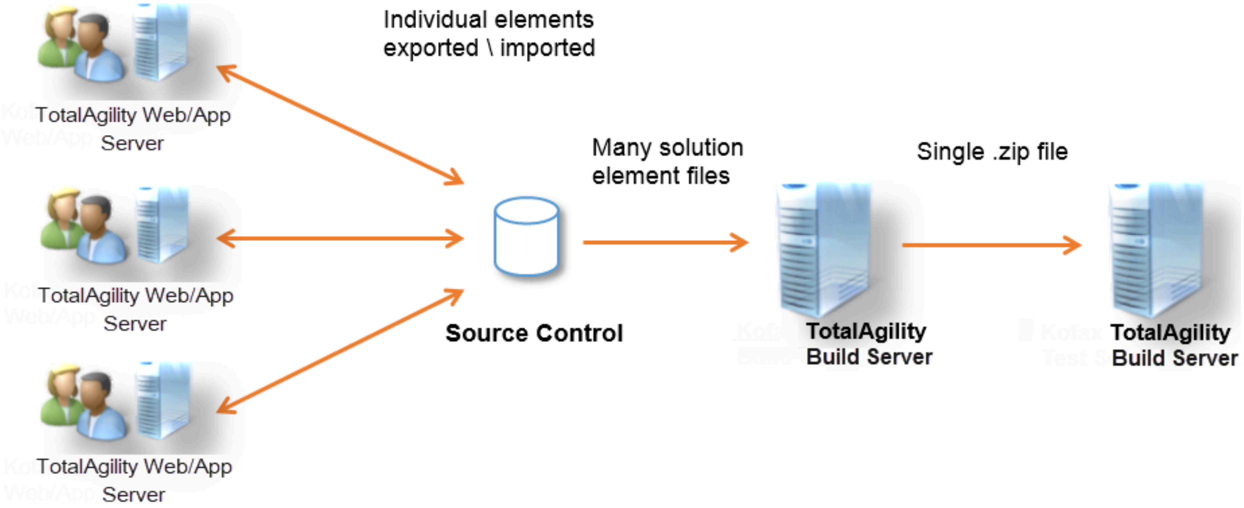
- Fewer development environments to maintain.

Disadvantages

- All activities occur on the server; therefore, a change that prevents an activity from functioning could affect and delay other developers.
- Only one version of custom .NET code can execute on the server, making it difficult to test without affecting other developers.
- All developers are tied to the availability of a single development server.

Standalone deployment environment

The development team installs Tungsten TotalAgility on an unlimited number of development machines, giving the developer greater flexibility in the changes to implement and test.


Individual elements exported \ imported
TotalAgility Web/App Server
TotalAgility Web/App Server
TotalAgility Web/App Server
Source Control
Many solution element files
TotalAgility Build Server
Single .zip file
TotalAgility Build Server

Apply the following process for managing and tracking changes made to the solution:

1. Use the product Export functionality to create a separate ZIP file for each item, such as process and form.
2. Place the ZIP files into a configuration management tool, such as Team Foundation Server.
3. Appoint a team member as Build Master to manage the following:
   - Updating ZIP files
   - Adding or updating global variables
   - Adding or updating the theme and CSS
   - Adding or updating the navigation menus and more
4. Developers send the file and elements to Build Master in the form of an exported ZIP file with just the changed elements in it, and a note indicating what has changed, to assist the Builder Master.
5. Each day developers check out the ZIP files from the configuration management file, implement the required changes, and check them in when complete.
6. Each day, at a minimum, the Build Master must do the following:
   - Import the latest ZIP files from the configuration management.
   - Update the package with any new items and export the package.
   - Alert developers and quality analysts that an updated and working package is available.

Promote new releases to new environments

When moving solutions from a development server to a test server or onto a production server, use the Packages functionality. Packages enable you to quickly export all items associated with your solution. The following features are available to make this process easy, ensuring that no item is overlooked:

- Automatically include items associated with a process.
- Automatically include items associated with a form.

- Automatically include items associated with a category.
- Include packages (a separate package may be created for items changed as part of a subsequent release of a solution).
- Compare the package with the items on the system.
- Search for items recently modified.
- Import on the target server: Update the value of global and server variables, such as connection strings, to the appropriate values on the target environment.

Assign access permissions on each package to prevent others from making unnecessary changes and causing issues.

Resources

Allow or deny access permissions and configure work allocation based on your requirements.

Access permissions

Manage the access permissions to grant and restrict resource access to the Designer and processes.

Designer

Restrict the access to the main areas of functionality within the TotalAgility Designer through system settings to tighten constraints on those who can modify processes, forms, or more importantly, system settings.

Allowing access to "everyone" could result in incorrect changes being applied, causing your system to behave unexpectedly. By default, only members of the Administrators group have access to these areas. For example, any resource that has not been granted appropriate permissions for the Designer will not be able to open the Designer to view processes, forms, and other areas.

See "Assign access permissions to different areas of TotalAgility" in the *Tungsten TotalAgility Designer Help* for more information about configuring the Designer access permissions.

Process

Restrict the access to highly sensitive processes using the maintenance access functionality within the process, so that changes to processes can be controlled and only made available for use once they have been thoroughly tested and approved. As only one resource can have maintenance access, we recommend that you configure a group for this purpose.

Refer to "Maintenance access" in the *Tungsten TotalAgility Designer Help* for more information about configuring process access permissions.

Activity allocation

Work can be allocated automatically as soon as it becomes pending, manually by a supervisor, or scheduled to be allocated at a specific time. We recommend automatic work allocation as it does not require any manual intervention.

Manual work allocation

If you require work to be specifically allocated to a resource by a supervisor before working on it, select the Allocate property on the activity. The activity does not become pending until the supervisor manually allocates it to a specific resource. See "Allocate work" in the *TotalAgility Workspace Help* for more information on manual allocation.

Scheduled work allocation

To automate the process of allocating work, use the "SYSTEM Perform Auto Work Allocation" process map available in the System category. This map can be scheduled to execute using the Job Scheduler at an appropriate interval and can be modified if required.

Automatic activity allocation

We recommend the automatic activity allocation because the appropriate work is allocated to the correct resource as soon as it becomes pending, thereby increasing the user’s productivity.

The following scenarios highlight the most applicable resource assignment feature to use.

Static versus Dynamic resource allocation

If you know in advance which resource group or person can perform the activity, use static assignment; otherwise, use dynamic assignment. For example, if an activity is related to personnel hiring, the HR resource group would perform that activity, and a static assignment would be recommended.

When configuring resources, use groups or roles (fixed or floating) to give more control over the runtime allocation.

Apply rules if the resource assignment is more complex as rules provide even greater flexibility. See "Assign resources by applying rules" in the *Tungsten TotalAgility Designer Help* for more details on assigning resources using rules.

Role versus a resource group

Use roles when the individual performing a task is unknown or is likely to change regularly. A role provides more flexibility than a resource group.

- A role does not require you to provide the name of a specific individual. At the time of configuration, you may not know the name of the actual person who will perform a task, but you may know the role required.
- The person assigned to a role may change on a job-to-job basis whereas resource groups are defined system-wide.
- Both resource groups and roles are associated with ALL versions of a business process. However, when role properties change, you do not need to re-release a process. This means the version number of the process map does not increase.

External resource

Use external resources in combination with floating roles for resources that are not part of the organization but are required to complete activities within the process (example: a loan applicant). These users can participate in the process but with limited access to the TotalAgility Workspace and no access to the TotalAgility Designer.

If you require the external resource to create jobs or cases or view the progress of a job or a case, use the limited user session in conjunction with the limited user license. This gives more access to non-TotalAgility users but is still restricted and managed.

Export resources

As resources are typically different between Development, Test, and Production, we recommend that you export groups only and not individuals. Where individuals need to be exported then you can do so by including members.

Building processes

A process is an orchestration of activities; it is not a system of record. Therefore, do not use process variables to store any information that is not required for decision making, searching, or passing onto nodes.

A case can incorporate several processes involving various departments using multiple sources of information. For example, processing an appeal could consist of a wide range of documents and forms, and numerous processes, such as registering an appeal, setting up a tribunal session, checking medical records, clearance of an appeal, and many more. These processes may run independently of one another, yet they are all related to one case.

TotalAgility supports multiple versions of the same process at any given time so a job can complete on the same version it was started on.

- Be careful when updating processes that are embedded or used as subjobs.

  Consider whether you want live parent jobs to have this change, as a parent job uses the latest version of the subjob or embedded process. If not, create a copy of the process and use that copy going forward.
- Group processes into logical categories, such as Invoice approval, PO processing, HR processing, Shared, and more.
- Use categories within categories if you wish to have one overall category for your solution.

Case versus process

A process is a highly repeatable straight-through workflow whereas a case is somewhat unpredictable in its path.

A process has a starting point, a defined path, and an end point. For example, you can clearly define a holiday request process up front and routinely execute it.

A case process consists of a base "case process" used to support the overall case, as well as several processes or fragments, all of which are linked for collecting and sharing case-specific information. This does not necessarily execute in a logical start-to-finish manner. It can be the responsibility of the Case Manager to determine the next steps to perform, or logic built into the case can automatically determine the next steps and create associated jobs based on fragments and other criteria. A consolidated case history is maintained across the entire case in chronological order.

For example, processing an Appeals case could consist of a wide range of documents or forms, as well as numerous processes such as registering an appeal, setting up a tribunal session, checking medical records, and clearance of an appeal. All these business processes may run independently of each other yet be related to a single Appeals case.

If you know the entire path from start to finish, and a definite set of steps must be completed in a set order, use a **process**. If the required steps need to be determined at runtime depending on certain criteria, use a **case**.

If you are still not sure whether to use a case or a process, use a process. It is possible to upgrade a process to a case or a fragment, but not vice versa.

Fragment versus process

Use a **fragment** if you need direct access to case data, milestones, states, events, and other details at both design time and runtime. A case fragment is similar to a process; it has all the attributes of a normal process, such as nodes, data, SLAs, and resources. However, a fragment is dependent on the case in which it is created and cannot be reused by other processes or cases.

- Ensure to align fragments with the correct version of the case process if necessary.
- If an updated fragment needs to be used in older versions of a case, ensure it is designed to only use artifacts that are available in the oldest version of the case. Trying to use artifacts that are not available can result in unexpected behavior or job suspension.
- Be aware; by default, new fragments are only compatible with the latest version of the case and there are no automatic checks that you can only use artifacts from the base version. This particularly applies when using fragments as sub-processes.

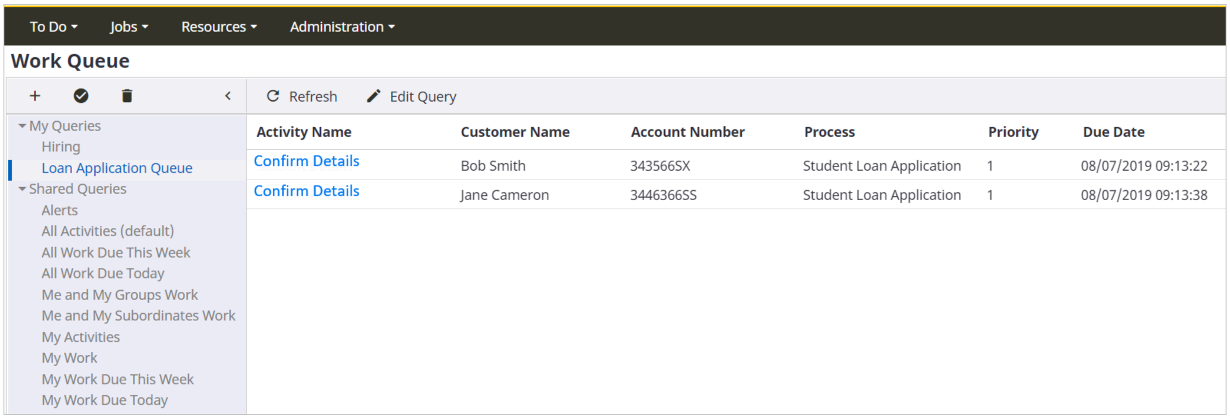
Use a **process** if it needs to be used independently from a specific case. As there is no direct access to case data, the data will need to be passed into the process or retrieved real time. Example: Send an email to the customer's preferred communication channel but record it as part of the case.

General practices

This section describes general practices that must be followed for optimal results.

- We recommend that you use work types for logical grouping of processes and metadata searching. Work types create a more informative work queue and job lists without the need to drill into each item to get access to key information. If you wish this information to be different throughout all fragments within a case, be sure to set the Scope property on the metadata to Yes.
- Where a work queue may contain many types, consider using supporting information to display additional information, this may particularly apply to mobile devices where on-screen real estate is very limited.
- Create a query for the work type. For example, a loan application gives flexibility so that the job information (customer name and account number) can be displayed and filtered within the query editor.

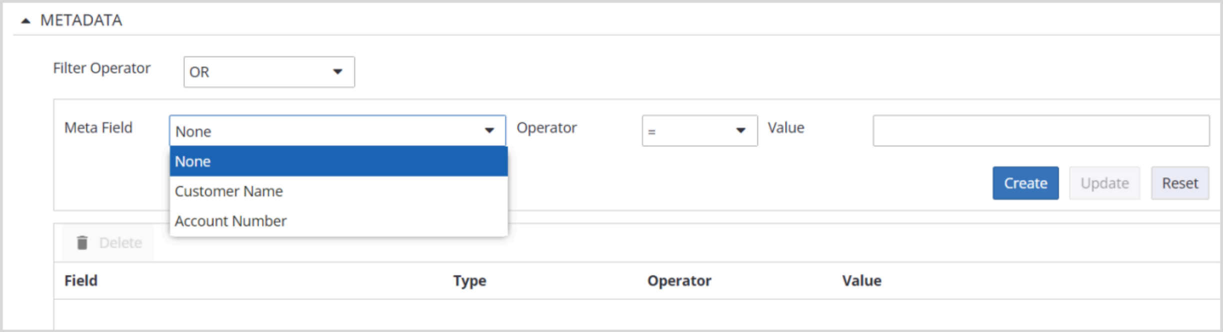
  Work queue



Work Queue

Activity Name	Customer Name	Account Number	Process	Priority	Due Date
Confirm Details	Bob Smith	343566SX	Student Loan Application	1	08/07/2019 09:13:22
Confirm Details	Jane Cameron	3446366SS	Student Loan Application	1	08/07/2019 09:13:38

Query editor - metadata filtering



Filter Operator: OR

Meta Field: None

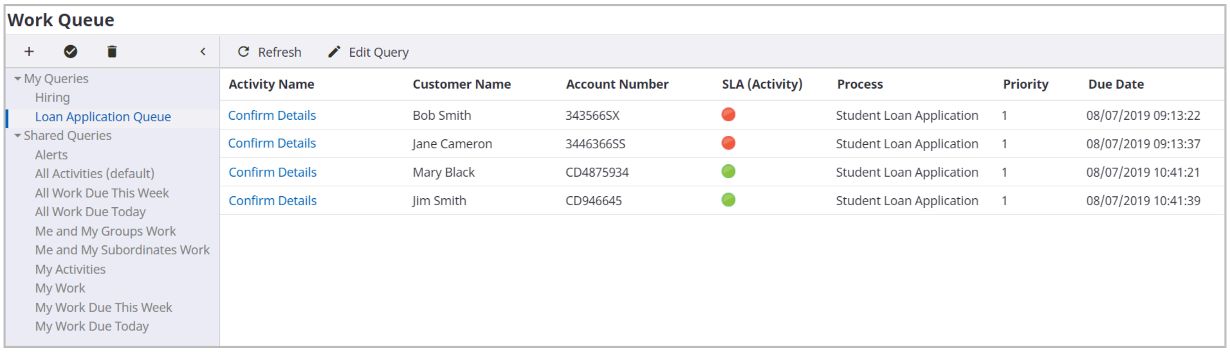
Operator: =

Value:

Field	Type	Operator	Value

- If you have service level agreements to adhere to, use the SLA functionality at both the process and activity levels. This will provide a more informative work queue and job list, providing a visual representation of when a job is at risk of not meeting its targets.

Work Queue - Loan Application Queue



Work Queue

Activity Name	Customer Name	Account Number	SLA (Activity)	Process	Priority	Due Date
Confirm Details	Bob Smith	343566SX	🔴	Student Loan Application	1	08/07/2019 09:13:22
Confirm Details	Jane Cameron	3446366SS	🔴	Student Loan Application	1	08/07/2019 09:13:37
Confirm Details	Mary Black	CD4875934	🟢	Student Loan Application	1	08/07/2019 10:41:21
Confirm Details	Jim Smith	CD946645	🟢	Student Loan Application	1	08/07/2019 10:41:39

For example, the SLA (Activity) column in Work Queue - Loan Application Queue displays that the loan applications for Bob Smith and Jane Cameron are not meeting the deadline.

- Be proactive before these activities become overdue so that you can take corrective action without much cost. Design your process to self-administer using exceptions and triggers. For example, configure a trigger to launch before the activity is due so that work can be reassigned, and raise an exception after the activity is due.

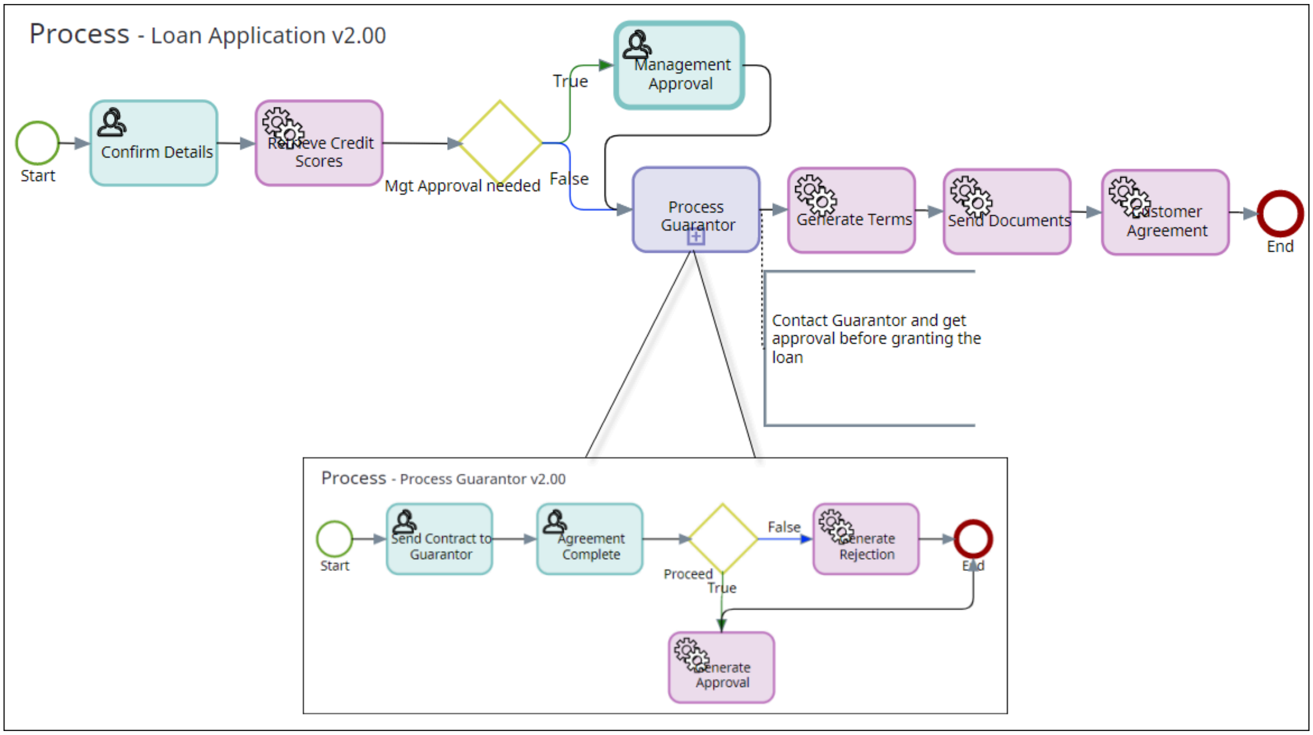
- If a process within a solution is required to provide a quick response, consider synchronous processing, completing an activity and progressing, creating a new job and progressing, and other such functionalities within the solution design.

Process design guidelines

Use the following guidelines when designing a process.

- Use clear display names for variables so they adequately reflect the use and purpose of the variable.
- Use subjobs and embedded processes to create logical groups of activities. Keep the flow of the process easy to understand to make maintenance easier. Use the rearrange option to display the process in a linear fashion, making it easier to read.
- Use clear display names for variables to reflect their use and purpose.
- Color code activities to make their intent clearer. For example, the following Loan Application process uses custom color coding for the manual activities that may hold up the process. Custom colors are also used for the integration points and the embedded process.

  The purpose of the embedded process is to process the guarantor. This processing can then be easily used within other banking processes, ensuring that any changes are isolated and require less testing.



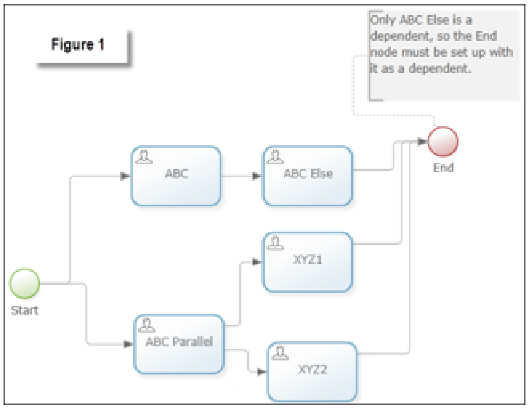
- Use annotations to provide additional context or explanation for the design.
- When modifying work types and adding and removing fields after they have been released into production, avoid deleting fields to add another with a different type. This can cause problems if jobs already exist as the data is held in the database as strings at a specific position and is converted when used. For example, if the Name (String) field is at position 1 and the Age (Numeric) field at position 2, and if several jobs are already created and you decide to delete

Name, there will be runtime issues; modification will fail because the system will try to convert the existing jobs which may have had alphabetic text, to a numeric value.

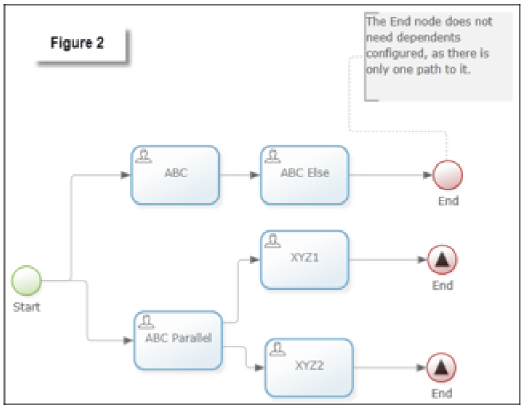
- Use different types of Start and End nodes as a visual aid to give more clarity to the process design.

> Not all end nodes end the current job; as a result, you can use them to end a path while other paths continue processing. See " Event types for an end node" in the *Tungsten TotalAgility Designer Help* for more information on these nodes.

- When designing a map that has non-dependent parallel paths, use non-completing end nodes to ensure your map is more readable. Avoid unnecessary synchronization points or dependent configuration. See the following figures for examples.



Only ABC Else is a dependent, so the End node must be set up with it as a dependent.



The End node does not need dependents configured, as there is only one path to it.

In Figure 2, the job is not completed when "XYZ1" and "XYZ2" activities are completed. The job is only completed when the "ABC Else" activity is completed. The dependents are not configured on the End node as was done in Figure 1 and multiple converging lines do not exist. This makes the map in Figure 2 more readable.

- Avoid naming a case fragment variable the same as a case variable. Keeping the same name causes issues when deleting the case variable, as the system may mistake this variable as being used within the fragment. It may also lead to confusion when viewing the map configuration, as there is no indicator within the map as to which variable is being used. At runtime, the scoping rules, process, case, and server will result in the process variable always being used.

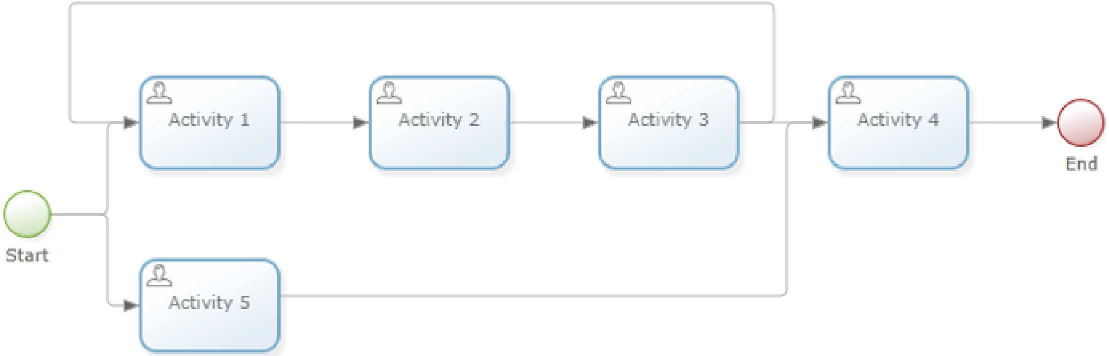
Process performance guidelines

Use the following guidelines to increase the process performance.

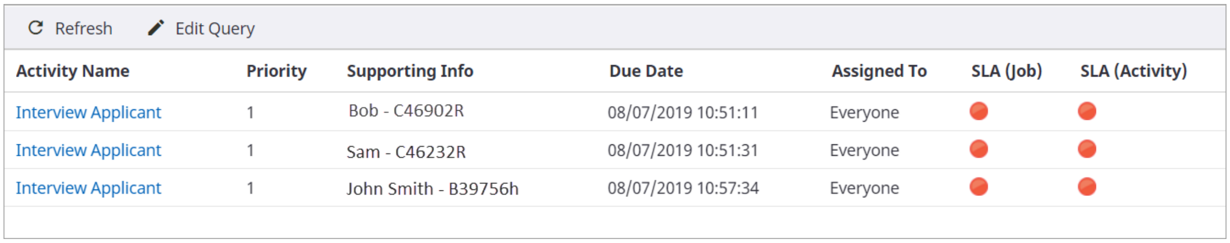
- Use process events for internal and external inter-process communication. Example: Processing the loan application can continue only once the loan agreement is signed.
- Use activity triggers and preconditions to pause a job and only use them for very short waits when you need to call external systems. See the topic "Create a process event" in the *Tungsten TotalAgility Designer Help* for more information on configuring and using process events.
- Business rules and synchronous processes are performance-oriented, hence we recommend you do not use Sleep nodes in Synchronous processes and Business Rules, as they impact the system performance. However, for the asynchronous process, you can still use the Sleep node to wait for the work to complete; this will not slow down the system performance.

- Long-running activities are canceled by TotalAgility if they go beyond the defined timeout period. However, under certain situations, some activities, such as the RPA robot may run longer. Therefore, before deploying, ensure that an appropriate timeout to specify how long an activity is expected to run is defined for the specific activity.

  Do not just change the system setting because that will result in activities being allowed to run and consume machine resources longer than they should and thus impact performance.
- If you do not require to track the job history for auditing purposes, clear the "Record history" setting within the process (Process properties>History, reporting and execution tab). This helps manage your database size and improve performance.
- Be careful of processes that repeat the same nodes (or set of nodes) with conditions and dependents. They can stop the map progress, waiting for dependents that never become active because of the conditions. Example: If Activity 3 is set as a dependent on Activity 4, Activity 4 may never become pending.



- Do not use Synchronization nodes unnecessarily. Synchronization nodes can merge and converge multiple paths in a process, adding load to the system and slowing down the database performance. Design processes with multiple paths into and out of the Synchronization nodes and use dependents if all paths must be complete before the Synchronization node can complete.
- If you wish to use a general work queue or job list form instead of work-type specific queues, use the "supporting info" functionality to display the job-related information.



Activity Name	Priority	Supporting Info	Due Date	Assigned To	SLA (Job)	SLA (Activity)
Interview Applicant	1	Bob - C46902R	08/07/2019 10:51:11	Everyone	●	●
Interview Applicant	1	Sam - C46232R	08/07/2019 10:51:31	Everyone	●	●
Interview Applicant	1	John Smith - B39756h	08/07/2019 10:57:34	Everyone	●	●

- It is recommended to use a single DataObject process variable containing multiple fields, instead of retrieving or updating multiple related simple process variables while taking or completing an activity.

Process testing

We recommend the following practices when testing processes:

- Test all paths of your processes before they are put into production.

- Use the out-of-the-box forms first to ensure the process executes as expected before creating the user interface for the solution.
- Use the restart functionality within the job viewer instead of creating a new job and running it through many steps to test a single part of your process. For example, if you make a change to a business rule, restart the process at the business rule node instead of creating a new job.

> In some instances, changes require either a new job or job upgrade.

- Understand the scope of your testing. A small change does not necessarily mean that you must retest the entire map.
- When using subprocesses or embedded processes, be aware that only the latest version is taken. Ensure that the process can be run and tested in isolation. For example, if the initialization data of a contract is changed, it may cause issues when the subprocess is consumed. If there is any need to align the process to a specific version of the subprocess, create a copy for the updates required and use the copy where needed. Alternatively, use cases and fragments so that the versions are aligned.
- If you are using a business rule, use the built-in business rules testing feature to test using various input values.

Set functional access

You can set a range of access types to control what functions can be performed on a live job.

By default, everyone can create, suspend, terminate, and restart a job, place the job on hold, view job details, change scan/VRS profiles, and change separation profiles.

By default, no one can create a customized version of the process for a live job.

We recommend that you assign access per function to more specific resources considering the skill and role of the resources. For example, consider which resources should be able to create jobs, which resources should be able to restart or terminate (ensuring those resources recognize the consequences of terminating or restarting a job), or which resources should be allowed to only view the details of the job.

Building forms

Use the following practices when building forms.

Suitability of using forms

The TotalAgility Form Designer is not intended to be a replacement for Visual Studio. However, it is intended to simplify the creation of forms using a drag-and-drop interface. When selecting forms as the intended development environment, it is important to understand what can be built using TotalAgility forms and its limitations.

- Keep forms simple and use the TotalAgility strengths. A specific strength of TotalAgility Form Designer is the automatic generation of forms used for core functionality such as Create new job/ case, Work queues, and activity progression with or without capture functionality.

- For solutions using capture-related functionality, use the out-of-the-box TotalAgility forms for the core capture-related activities. These forms can significantly reduce development time.

Design guidelines

Use the following guidelines when designing a form.

- Ensure your form is readable by using cells and columns to create the required layout instead of relying on the margins and hidden controls. Each form should be easy to understand, and a new member of the team should be able to follow the logic and easily maintain the form.
- Use clear display names for controls and actions that reflect their use and purpose; you cannot change the names once created.
- To reduce maintenance complexity and cost, do the following:
  - Ensure that no event fires more than 10-20 actions.
  - Events do not contain excessive conditional logic.
  - Keep the number of controls on a form to a minimum.
- All forms are set to use the default menu and header. Update or remove the menu and header where not needed. For example:
  - The default header has links for unread resource notes and makes an API call. If you are not using the resource notes, change the header or update the existing one.
  - Remove the menus from the activity-based forms to prevent the user from navigating away from the page without canceling the activity. Updating the menu and header is particularly relevant to capture activity, as it also retains the document locks.
  - Removing the menus also applies to forms that contain logon or logoff capabilities. Not updating these may provide access to data or functionality without a valid session.
- It is common for desktop and phone forms to have significantly different navigation and form flow. Therefore, it does not make sense to use the same form setting and have a desktop, phone, and tablet versions within the same form. Decide the structure of your solution early and then select the appropriate design factor.
- Use multi-view activity forms when you need to view other systems or data needed to complete an activity.
- Use current view in phone forms, especially for MFP devices using MFP forms, to significantly improve redirects. This helps in avoiding the lack of resources on the device reloading the Extjs libraries and results in faster load times. While still applicable to other mobile devices, the same level of improvement is not seen in them.

Be aware that TotalAgility has no built-in support for compensating transactions. If required, create them as part of the implementation project.

Combine capture with non-capture

When enhancing capture forms with non-capture data, follow these practices.

- If possible, keep non-capture data in the activity or initialization screens; otherwise, some unexpected behavior may result. For example, mandatory fields on hidden tabs may not be visible to users.
- If fields are needed against the document, consider defining them against the document instead of defining them just on a form. For example, adding a non-capture field to a capture form and

making it mandatory can create unexpected results when validating a folder, as the document form is not loaded and therefore the expected validation would not be performed.

Form regeneration

TotalAgility supports multiple versions of the same process at any given time so a job can complete on the same version it was started on.

Use the form regeneration functionality to support different versions of processes and different versions of the associated forms.

There are various options within the regenerate functionality that will result in a different output.

- Whether to create a new version of the form or overwrite: Create a new form when you have live jobs that require the current version of the form to run successfully.
- Whether to save or release the new form: Release the form only if it is available for use.
  - The corresponding process has been released.
  - The form does not need any further modifications.
  - New jobs can use it.

Reuse forms/navigation/headers

Reuse forms, navigation, and headers to avoid rework, and easier maintenance.

- For activities with minimal fields, or documents and folders with a small number of fields, you need not always generate a form. Before you generate a form, review what is required on the form and if any customization or custom logic is needed. Once you generate a form, the system automatically creates the user interface for you at runtime. This reduces the number of forms, and maintenance required subsequently.
- Do not assume the Workspace navigation as the default for your solution. The Workspace and its default navigation provide immediate access to everything including the administration features that is not required normally. Design the correct navigation required for your solution context.
- Consider removing the navigation menu from the Take Activity forms to ensure users complete or cancel activities instead of browsing away and leaving the activity in a taken state.
- If the same functionality, such as work queues and administration pages, is required in multiple solutions, create these forms once and share them between solutions. You can style them differently using the theme associated with the site.

Security

Consider who should have access to forms and menu items in their entirety and whether all controls on a form should be available to all users.

Security is available on forms and navigation menu items to allow or deny access to specific resources. By default, everyone has access.

For a more granular level of security, use the "Security tokens" feature. Assign tokens to individual controls or cells on a form, and only resources who belong to a role with these tokens can see the controls.

ⓘ There is some overhead involved in rendering forms that have security tokens, as extra calls need to be made to determine the roles of the current resource before determining the controls to display.

Forms maintainability

Consider future maintainability when creating or modifying a form, as this is a key factor for the success of any solution.

Reuse forms across activities so long as the contract (input and output) of the activity, such as number, name, and types of all inputs and outputs, are the same. This reduces the number of forms and subsequent maintenance. Set the following process or activity properties to use the same form.

Job creation form (Process properties > Access tab)

Associated file path (Activity properties > Configuration tab)

Form loading events

When using the "Loaded", "BeforeRender" and "AfterRender" events, make sure that they are fired in the following order:

1. Loaded
2. BeforeRender
3. AfterRender

The **Loaded** event is the most commonly used event. Actions, such as Same Page, .NET and DB Query that initialize controls or form variables are typically added to this control. Form controls cannot not be loaded into the DOM at this point, so do not place actions that attempt to access form controls through DOM (example: Javascript actions).

The **BeforeRender** event is fired at the point at which the form is about to be rendered. For example, the BeforeRender event can be used to inject CSS into the head of the document prior to rendering. As per the Loaded event, the Javascript actions executed here should not attempt to access form controls through DOM as those controls will not be available at this point. See Access DOM elements through Javascript actions.

The **AfterRender** event can be used to access DOM elements since the form is guaranteed to have been fully loaded into the DOM. This is specifically useful when adding EventListeners to DOM elements to achieve some non-standard functionality.

ⓘ The Capture activity forms operate differently from other form types in how they render. This is due to the ThinClient control, which requires that the onLoad actions complete before it can fully render itself. So even though the BeforeRender or AfterRender events may have triggered on the containing form, the form itself may not actually have been rendered to the required state.

Access DOM elements through Javascript actions

The control must be fully rendered for accessing the control's value in a Javascript action by means of a DOM query.

For example, take the following line of a Javascript action:

```
var controlValue = document.getElementsByName("textbox1")[0].value;
```

If we associate this Javascript action with either of the "Loaded" or "BeforeRender" events, it fails as the form is not rendered fully and the DOM element is not available.

However, by associating the action with the "AfterRender" event, the code executes as expected as the form is fully rendered. Associating the action with any control event, such as "TextChanged" or "Clicked" also succeeds as the form is fully rendered.

Associate actions other than Javascript actions

Non Javascript actions, such as Same Page or .NET actions are not subject to the same limitation as Javascript actions, since these actions do not attempt to directly access DOM to get or set control data. For example, a Same Page action will successfully map data from one control to another irrespective of whether it is associated with a "Loaded", "BeforeRender" or "AfterRender" event.

The server-side actions, such as .NET actions execute synchronously (one after the other) on the server when they are defined sequentially for an event. However, when multiple events are raised, each set of server side actions are executed asynchronously from the client even though the actions themselves execute synchronously on the server. Therefore, be careful when associating server-side actions with "Loaded", "BeforeRender" and "AfterRender" events, as there is no guarantee that the server-side actions associated with one event will have completed before the actions associated with another event. The order of completion may not be as expected.

Test forms

Test your forms thoroughly.

- Use the "Restart at" functionality to minimize the number of jobs created.
- Test business rules and other integration items in isolation before integrating them into your form.
- When multiple actions are involved, deactivate or activate some actions to locate the issue and isolate the problem.

In the absence of any release or development versions of a form, be aware that continuous releasing of a form for test purposes increases the final version number within the solution.

Prevent skipping validation activity from the work queue

From the TotalAgility Work queue form, you can auto-complete an activity without taking the activity at all. To do so, on the "Actions" column of the activity, click "Complete Activity". The Validation activity is completed even if the job still contains invalid documents.

To prevent users from auto-completing an activity, and to ensure the activity is actually worked on before being completed, disable the auto-complete feature from the work queue. To do so, open the Work queue form. In the properties panel of the Work queue control, on the "Configuration" tab, under "Actions", ensure that "Allow auto complete" is clear.

Business rules

A business rule is a means of implementing complex business logic without the need for custom code. Its main purpose is to determine a result, based on the input provided. For example, determine the interest rate on a loan depending on the agreed terms and conditions. Business rules cannot contain manual activities as business rules are intended for straight through processing and should not be used for updating systems of record and other such activities. If you require this functionality, use a synchronous process.

Follow the process design guidelines when designing a business rule, avoid unnecessary inputs and outputs, and use subrules where necessary.

Business rule maintainability

Ensure the contract (inputs and outputs) is not changed after deployment to production; otherwise, the business rule will break where it is used. If you need to change the business rule’s contract, ensure the rule is no longer used in production, or create a copy of the rule and use it to create a new rule. Delete the old rule if not using it.

Business rule testing

You can use the test functionality within the properties of a business rule to provide sample input and view the resulting output. If the output is not as you expected, you can manually walk through the flow to determine where it went wrong, or you could add tracing by using an output variable and updating its value at various points using the expression nodes.

Be aware that adding the tracing will require the rule to be released, thus increasing the version number of the business rule.

See Process testing for more information.

Handle import events

When a message is imported using the storage mode or direct mode, for a successful import, the message should be deleted/confirmed from the source. In some cases, the message is not deleted/confirmed from the source, and it can be imported again and may result in a duplicate import.

You can use process maps that can take appropriate paths based on different import events which are triggered during the document import. Based on these events, a process map can avoid any duplicate import scenario. You can view the following events for a job in Tungsten TotalAgility Workspace.

- **IMPORTRUN**: This event is triggered when the confirm/delete request is successful from the source system.
- **IMPORTDUPLICATE**: This event is triggered for one of the following scenarios and may result in duplicate document import:
  - If the source system does not receive the confirm/delete request due to errors.
  - If the source system responds that the message is already confirmed. This means that the message is already imported by another import source.

> ⓘ An error message is configured in the ErrorMessage variable.

If the source system does not receive a confirm/delete request due to errors, such as network failure, and no event (either IMPORTRUN or IMPORTDUPLICATE) is triggered for a configured timeout, the process moves to the next node in the map. Default: 30 minutes

The sample process maps (Import Connector Event Handler Package.zip) are available in the TotalAgility installation folder. The default location is:

```
C:\Program Files\Tungsten\TotalAgility\Sample Processes
```

You can import this ZIP file to the TotalAgility Designer, and then embed the ImportConnectorEventHandler process map in your existing process map or can create a new process map and embed the same in your process map.

Job Upgrader

Use the Job Upgrader tool if you need a change to a process design to take immediate effect across all new and live jobs. For example, a legislation change requires a change in SLA.

Use the Job Upgrader only when necessary. Be careful when you use it, as it may result in some unexpected behavior. For example, if a new variable is used within a new activity, any upgraded jobs may only have the default value. Or if you remove document type metadata fields from the document set that are referenced and try to upgrade jobs, it may result in unexpected behavior.

Apply the same rules to the upgrade as you would apply to any software upgrade. For example, back up the database and do appropriate regression testing. For the regression testing, use a backup of the live database and use the actual jobs that are to be upgraded as your test data.

Data management

Entities provide a lightweight mechanism to define and store data in TotalAgility, such as the details of a company, a customer, addresses, and more. They are suitable for non-technical users, or when you do not have database or database skills.

They do not provide the same level of scalability or security provided by an actual database model. It is not possible to move the actual instance data between systems.

You can only perform a basic search but cannot report on data.

This feature is not suited for processing complex data structures that require strong data integrity, such as views across multiple tables, foreign keys, and intensive searching.

Distributed upgradability

When building a solution that will be distributed to many customers, design your solution considering upgradability so that you are able to send upgraded versions (example: 2, 3, 4) of your solution to multiple customers.

Every package should be made up of two parts:

- Core: Protected items that the customer cannot change.
- Custom: Items the customer may change.

A new customer should import a package containing both packages.

You can update the Core package and send it to customers based on your release cycle, as you are in complete control of this.

The Custom package requires customers to decide whether they want these or not. Always take new items as they are likely needed based on possible core changes. If items are modified, again it is the customer's decision whether they need the updated items. However certain rules must be obeyed. For example, customization cannot change the contract (initialization data) between any core items and custom items. This also relates to forms, such as Create New Job and Activity forms.

Design your processes with customization points.

Chapter 8

Troubleshooting

This chapter describes several issues related to business process management, Capture, Transformation Server, and VRS, and provides the ways to avoid or overcome those issues.

For installation troubleshooting, refer to the *Tungsten TotalAgility Installation Guide*.

Business process management

This section describes issues related to forms, processes, business rules and performance. It also describes the general issues and possible ways to avoid or overcome those issues.

Forms

- If a form does not display as expected, use "preview" to troubleshoot, or deploy the form and test its function in the deployed environment.
- If a form does not function as expected, for example, controls do not populate correctly or actions do not perform as expected, we recommend that you disable all other actions that do not depend on the action being debugged to isolate the behavior.
- If you wish to see data that is currently not on the screen, such as form variables, create a debug panel and use the Same Page actions to populate controls with the required data. Once the troubleshooting is complete, remove the panel.

Processes

- If requiring a **synchronous process** as part of a solution, first create the process as asynchronous to ensure the behavior is correct. Once the synchronous process is created, do the following to debug the process:
  - Check the event log for additional error information.
  - Turn the Record history on to view the path taken through the job viewer and to see the final values of the variables.
  - If you need to break this down further, add decisions or branching rules (XOR) with end nodes at the various breakpoints to view the variable values at that point.
  - Once the problem is isolated, copy the node into a test process for further isolated testing.
- To debug an **asynchronous process**, do the following:
  - Check the job notes for additional error information.
  - Turn the Record history on to view the path taken through the job viewer and to see the final values of the variables.

    - Add manual activities to simulate breakpoints, and to interrogate and set variable values.
    - Use the "Restart at" functionality to jump to the breakpoint.
- When work does not appear on the work queue as expected, use the Job Viewer within the Workspace to determine the location of the current job in the process and the resource assigned to the activity. Also, ensure the following:
    - The activity has not been removed because it is on a non-dependent path, paying particular attention to any loops.
    - The members are as you expected if the activity is assigned to a group or a role.
    - There are no preconditions stopping the activity from becoming pending.
    - There are no dependents stopping the activity from becoming pending, paying particular attention to any loops because each dependent path must be executed the same number of times.
    - The activity is not already taken; you can also see this through the Reset Taken Activity screen of the Workspace.
    - You are not an excluded resource.
    - Any exit rules have not been met.

Business rules

Test the business rule within the TotalAgility Designer with sample input and output. If the issue persists, see the Synchronous section of Processes for further troubleshooting.

Performance

If the general performance of a process is not good, use the process history to view the duration to determine if the execution is slow. Use the Time Pending (in secs) parameter within the database for Job History to determine if polling leaves the activity pending for longer than expected.

If system tasks, such as job evaluation, retention policies, archiving, or monitoring are not executing as expected, ensure that the execution interval is as expected, and the core worker service is started.

General

If the automatic activities are not being performed, yet the core worker is running and there are no errors being written to the logs, ensure that there are no looping activities taking precedence over other scheduled activities.

Capture

If documents classify or extract differently in the Transformation Designer and at runtime in the Transformation Server, it is most likely due to the actual images being classified or extracted being different.

To analyze this, do the following:

1. Place a temporary dummy activity into your process just before classification or extraction so the workflow stops.
2. Open the Repository Browser and find the document.
3. Export the document to disk using the context menu.

If you load this document in the Transformation Designer, you can analyze the difference between this document and your original document.

Transformation Server

To analyze the failure of the Transformation Server to pick up activities, ensure the following:

- The Transformation Server is running.
- The account specified in the installation for the Transformation Server service has “run as a service” set in Windows local policy.
- “EnableSynchronousCalls” is set to false in the Transformation Server configuration file if you expect it to process non-push activities.
- The capture activities do not have an unmet precondition defined in the Process Designer.
- Sometimes the Transformation Server goes into an interval of polling when it does not find more tasks. You can lower the length of that interval by configuring the "PollingTimeout" in the configuration file.

  ⓘ The production systems do not support Polling Timeout.

VRS

After importing a process or package containing the Scan/VRS profile, you may receive the following error while trying to scan a new job:

The selected VRS Profiles(s) could not be found. In the event you require technical assistance please reference action: `ScanControlLoadBatchAction`.

If you receive this error, restart your IIS to ensure the imported Scan/VRS profile settings are propagated to the client.

Chapter 9

PDF handling

In Tungsten TotalAgility, you can import the PDF documents and process them through Capture workflows such as Document conversion, Image processing, Classification, Extraction, Document Review, Validation, and Verification.

Document conversion uses newer libraries therefore we recommend that you use the Document conversion activity to convert the files to TIFF, as it is more efficient.

Refer to the *Tungsten TotalAgility Designer Help* for more information on Document conversion activity.

Use the PDF generation activity to convert the image files (GIF, JPEG, PNG, HEIF, HEIC, BMP, JFIF) to PDF.

Image processing is still in the product for legacy reasons. If using Image processing, we recommend the following best practices.

- When using the Image processing activity to process the PDF documents, we recommend that you disable any processing algorithms that can alter auto crop, auto rotate, deskew, and other settings related to page dimensions.

  Disabling processing algorithms is important if the PDF text layers are being used for extraction, or if the image processing leads to dimensional differences with PDF documents used in the Transformation Designer.
- Place the Image processing activity in a process as early as possible, especially before any Transformation and Validation activities. This is to ensure that Online learning works properly, as Online learning needs to occur on the same image as Transformation and Validation.

Use PDF generation to recreate the PDF document if PDF is needed in the business process.

Chapter 10

Extraction and Classification Group design

This chapter describes the best practices for a classification group and an extraction group design.

Classification and extraction groups

Classification groups and extraction groups need to be initially created in the TotalAgility Designer.

Create the extraction groups first, as the classification groups require at least one extraction group before they can be saved or released.

Once the classification and extraction groups are created, you can partly edit them in the Transformation Designer, and partly in the TotalAgility Designer.

Shared Projects

Creating a new project in the Transformation Designer creates a shared project. The same outcome occurs when a Tungsten Transformation project is imported into the Transformation Designer through the file system.

In a shared project, the document classification and data extraction can be defined in one Transformation Designer project. This type of project does not have a separate classification group or a separate extraction group.

ⓘ Rearranging the project structure for a shared project, a classification group, or an extraction group, must be done in the TotalAgility Designer.

Classification and extraction groups versus shared projects

The biggest advantage of working with classification groups and extraction groups as opposed to shared projects is the ability for project designers to work on a different classification group or extraction group simultaneously.

For example, a classification group (CG) includes two separate extraction groups.

1. EG_AB: For processing document types A and B
2. EG_C: For processing document type C

This scenario enables three project designers to work simultaneously on the project. For example:

- Project designer1 works on the classification of the three different types of documents
- Project designer2 works on the extraction of document type A and B
- Project designer3 works on the extraction of document type C

However, a disadvantage of using groups is that when you alter the project structure in a classification group in the TotalAgility Designer, any linked extraction group projects will also need their project structure altered to match the classification group. Since you cannot edit them at the same time, you must first edit the classification group, close it and then open the extraction group in the TotalAgility Designer. The extraction group classes can then be added or deleted, or existing classes can be moved within the project structure by using the class Parent property.

Additionally, if using multiple extraction groups, it may be necessary to duplicate WinWrap scripts across different groups.

Regardless of whether you are using a shared project or a combination of classification group and extraction groups, formatting, validation configuration, and validation form design need to be configured in the TotalAgility Designer.

Fields

The following sections describe best practices for fields.

Strong naming conventions

Use strong naming conventions when naming fields. The name should easily identify the field. For example, the names `PatientTelephoneNumber` and `InsuranceTelephoneNumber` are better suited than `PatTelNum` and `InsTelNum` for identifying the purpose and expected values for a field.

Since the names of fields are also used as variable names in the Transformation Designer WinWrap scripts, it is important that the names reflect their purpose.

If a project is passed on, a new designer may not understand the naming of the fields and may have difficulties in script or in mapping fields to formatting and validation rules in the TotalAgility Designer; strong and descriptive names minimize this problem. For more information on name conventions, see "Name conventions" in the *Tungsten TotalAgility Designer Help*.

Sequence and field groups

For each class, try to create fields in a logical sequence and make the list complete, including auxiliary or dummy fields, before saving or releasing.

Once a project is saved or released, any new fields are added to the field groups of that class. For example, for ClassA, creating Field1 and Field2 and then saving or releasing the project will add Field1 and Field2 to FieldGroup1 in that class.

In the Transformation Designer, if Field1a is added and moved to a position so that the sequence is Field1, Field1a, and Field2, after saving or releasing the project, Field1a is added to FieldGroup2. The next time the project is opened in the Transformation Designer, the sequence will be Field1, Field2 and Field1a.

The fields in the Transformation Designer are displayed in the sequence they appear in the field groups in TotalAgility Designer. As you cannot move the fields between field groups in the TotalAgility Designer, you must plan the fields and their sequence before creating them in TotalAgility.

Adding and removing document fields

While adding or removing fields from a document type, regenerate and rerelease the associated document form. If there are no active documents meaning the documents cannot be accessed by a capture activity, use the option to overwrite the existing form while regenerating the form.

If there are active documents of the updated type, use the option to create a new form when regenerating the form. In this case, the new form is used for newly created documents of the updated document type, and the previous form is used for the existing active documents in the system.

Chapter 11

Validation and formatter implementation

This chapter describes best practices for formatting and validation implementation.

Formatting

You can create formatting rules by applying formatting methods to a field in the Transformation Designer, but you can define formatting of a field in the TotalAgility Designer.

The formatting and validation methods are available in Transformation Designer because some locators, such as Invoice Group use these methods for extraction purposes. However, it may be necessary to use the date formatter in Transformation Designer for document types, such as invoices, because the date formatter in the TotalAgility extraction group does not have the month's replacement dictionary functionality. If you try to format a date, such as *January 13, 2016*, the TotalAgility date formatter fails to format it.

In some scenarios, it is also advantageous to use a formatting rule in the Transformation Designer to clean up the extracted value, before it is passed on to the TotalAgility extraction group for formatting and validation.

In the Transformation Designer, you can test the extraction of a document with or without formatting and validation.

- Testing without formatting and validation is helpful if you want to work on and improve extraction results.
- Testing with formatting and validation is important for benchmarking. The golden files used for benchmarking are typically formatted and validated. Use formatting and validation when benchmarking those golden files.

Field formatters

The field formatters in TotalAgility Designer (Capture>Field formatters) include three standard formatting methods: Date, Amount and Percentage, and one non-standard formatting method: Business rule.

A business rule is a TotalAgility process that has a Start and an End node with one or more *activities* in between. A Formatting Business rule has several input and output variables that are required. These variables are case-sensitive and must be defined in a particular order for the business rule to work.

The necessary syntax of the variables:

- Input variables (Name, type)
  - `FieldText, string`
- Output variables
  - `IsValid, boolean`
  - `ErrorMessage, string`
  - `FormattedText, string`
  - `FormattedValue, string`

There are many types of activities available. You can use the *Expression* activity to format a field value and configure several *Set Variables*, such as `Uppercase`, `Lowercase`, `Trim`, `Left`, `Mid`, `Right`, and `Replace`, in the expression.

C# activity and Visual Studio C# editor

The C# activity provides greater testing capabilities and has a better overview when compared to a business rule.

> ℹ The Project Designer must have some basic knowledge of Visual Studio C# to work on this activity.

You can edit the C# code in the Configuration tab of the C# activity in TotalAgility Designer. This code box in the Configuration tab is small, and offers no syntax highlighting, IntelliSense, and testing capabilities. We recommend that you use the C# editor in Visual Studio to create, edit and test your code.

The C# activity coding window provides some basic skeleton code that helps you get started. This code contains the `ScriptParameter.sp` parameter. The same parameter needs to be used in C#. To do this, you must reference the *Agility.Server.Scripting.dll* library in C#. This DLL file is typically available at the following location (default):

- `C:\Program Files\Tungsten\TotalAgility\CoreWorkerService\Agility.Server.Scripting.dll`

Example: Social Security Number

The Social Security Number in the USA has the format ddd-dd-dddd, where d is a digit. You can use the following C# code to format a field that extracts a Social Security Number:

```
using System;
using Agility.Server.Scripting.ScriptAssembly;
namespace TAformatter
{
    public class SocialSecurityNumber
    {
        [StartMethodAttribute()]
        public static void Method1(ScriptParameters sp)
        {
            string fieldText = sp.InputVariables["FieldText"].ToString();
            string formattedText = fieldText.Replace("-", "");
            formattedText = formattedText.Replace(" ", "");
            var num = 0;
            var isNumeric = Int32.TryParse(formattedText, out num);
            //check the length for 9 digits and the value to be numerical
            if (formattedText.Length == 9 && isNumeric)
            {
                formattedText = formattedText.Substring(0, 3) + "-" +
 formattedText.Substring(3, 2) + "-" + formattedText.Substring(5, 4);
                sp.OutputVariables["[FormattedText]"] = formattedText;
                sp.OutputVariables["[IsValid]"] = true;
            }
            else
            {
                sp.OutputVariables["[FormattedText]"] = formattedText;
                sp.OutputVariables["[ErrorMessage]"] = "SSN Value is not a valid";
                sp.OutputVariables["[IsValid]"] = false;
            }
        }
    }
}
```

Once the code is tested positive, copy the code and paste it to the code box in the Configuration tab of the C# activity properties. Validate the code and then test it, before saving and releasing the business rule.

Validation

The TotalAgility Designer has the following standard types of validation methods at the class level in an extraction group: Date, Regular Expression and Standard. These methods can only be used for single field validation rules

"Business rule" is a non-standard validation method that can be used for single field validation rules, but more commonly for multi-field validation rules.

A business rule is a TotalAgility process that has a Start and an End node with one or more *activities* in between. A Validation business rule has several input and output variables that are required. The input variables are the fields used in the multi-field validation rule. If you are only using a single field, only one variable is required. There are two output variables that are case-sensitive and must have the following sequence for the business rule to work.

- Input variables (Name, type)
  - `<FieldName_1>, string`
  - `<FieldName_2>, string`
  - .
  - .
  - .
  - `<FieldName_n>, string`
- Output variables
  - `IsValid, boolean`
  - `ErrorMessage, string`

Use the C# activity Configuration tab in TotalAgility Designer if the Business Rule is a simple and manageable process and does not require too many activities. For example, use the C# activity if the process can be viewed on one screen without having to scroll or zoom out as in the following scenario: Amount1 + Amount = Total.

This can be managed with one decision node and two expression nodes.

Use the Visual Studio C# editor, if the process is complex and requires many activities. For example, use the Visual Studio C# editor if the process requires scrolling and zooming out.

If the logic of activity is very complex and difficult to understand, create a custom DLL and use the DLL through the TotalAgility store in a .NET activity.

Example: International Bank Account Number (IBAN)

The logic for validating the IBAN in a business process would involve many activities and would quickly become cluttered.

The C# activities code box presents some basic skeleton code that helps to get started. This code has the ScriptParameter sp parameter. The same parameter needs to be used in C#. To do this, reference the Agility.Server.Scripting.dll library in C#. This dll is available at: `\\TotalAgility\CoreWorkerService`

The following C# code is easy to read and maintain. It can be developed and tested in the Visual Studio C# editor.

```
namespace StandardCaptureValidators
{
    using System;
    using Agility.Server.Scripting.ScriptAssembly;
    public class Iban
    {
.net activity</param>
        [StartMethod]
        public void ValidateIban(ScriptParameters scriptParameters)
        {
            var iban = scriptParameters.InputVariables["Iban"].ToString();
            if (this.ValidateGermanIban(iban))
            {
              scriptParameters.OutputVariables["[IsValid]"] = true;
            }
            else
            {
              scriptParameters.OutputVariables["[ErrorMessage]"] = "IBAN is not valid";
              scriptParameters.OutputVariables["[IsValid]"] = false;
            }
        }
        /// <summary>Validate German Bank Account Number</summary>
        /// <param name="iban">The International Bank Account Number</param>
        /// <returns>True, if it is a valid German Iban number</returns>
        public bool ValidateGermanIban(string iban)
        {
          // D=13 & E=14 & 00 -> 131400
          var numericalCountryCode = "131400";
          var checkSum = iban.Substring(2, 2);
          var calculatedCheckSum = Convert.ToString(98 -
 this.CalculateIbanCheckNumber(iban, numericalCountryCode));
          return checkSum.Equals(calculatedCheckSum);
        }
        /// <summary>Calculate the Internation Bank Account Number checksum</summary>
        /// <param name="iban">The International Bank Account Number</param>
        /// <param name="numericalCountryCode">The Country Code coming</param>
        /// <returns>The check number</returns>
        private int CalculateIbanCheckNumber(string iban, string numericalCountryCode)
        {
          var temp = iban.Substring(4) + numericalCountryCode;
          var part1 = Convert.ToString((Convert.ToInt32(temp.Substring(0, 6)) % 97));
          var part2 = Convert.ToString((Convert.ToInt32((part1 + temp.Substring(6, 6)))
 % 97));
          var part3 = Convert.ToString((Convert.ToInt32((part2 + temp.Substring(12,
 6))) % 97));
          return Convert.ToInt32((part3 + temp.Substring(18))) % 97;
        }
    }
}
```

Capture table input in a business rule

You can validate a table in addition to validating table cells. For example, mark an empty table invalid to prevent the validation activity from completing.

A validation business rule accepts both a table column and an entire table (table field) as input.

If the validation rules are intended for the table field and individual columns within the table, include all the logic within a single multi-field validation business rule. Do not apply single-field validation rules to individual table columns, and a separate multi-field validation business rule where the same table is used as an input.

If a table field is set to valid by a multi-field validation business rule, all rows in that table are marked as valid. Similarly, if a table field is set to invalid by a multi-field validation business rule, all rows in that table are marked as invalid. So even though a cell is marked invalid by a single-field validation rule on a table column, it can be overridden by a multi-field validation business rule that sets the table to valid. To avoid this condition, include all validation logic for a table inside a single multi-field validation business rule.

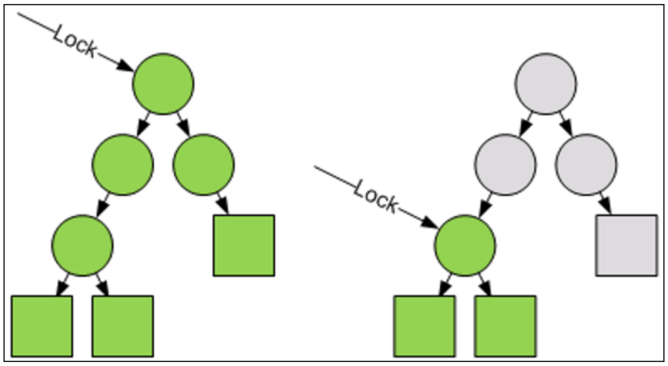
Do not follow this rule, if the multi-field validation business rule is not using a table field as input, or if a single-field validation rule is not assigned to any of the table columns.

Chapter 12

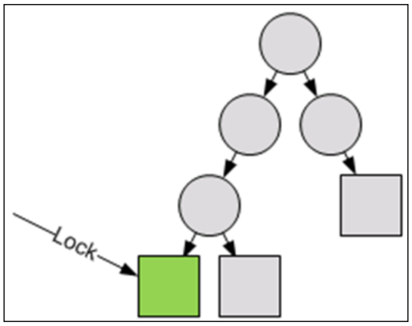
Folder and document locks

It is important to understand the locking behavior for documents and folders when designing your capture processes. Locks can occur at the document level or folder level.

A *folder exclusive lock* allows users to change any object within the folder hierarchy including both folders and documents. The lock state is propagated down to the hierarchy. So, if you locked a folder, you could change any object within that folder.



A *document exclusive lock* grants the user update rights for the locked document. However, it does not allow you to delete the document, split or move it to a different folder. To do this, you need to obtain the lock on the folder that contains the document.



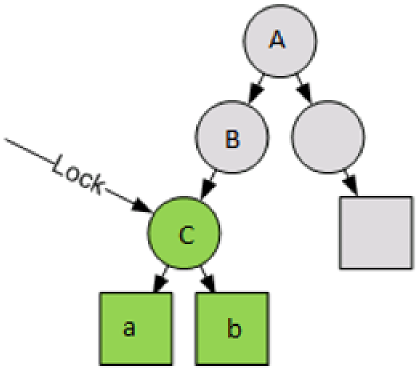
The Scan Create New Job (SCNJ) creates a folder and locks it when the form is loaded. Unattended and attended Capture activities also require locks.

Take care to avoid locking problems in the following scenarios.

Deleting objects

Deleting an object requires an exclusive lock of the folder directly above the object in the hierarchy.

In the following locking scenario, document "a" or "b" can be deleted but folder "C" cannot be deleted.



To delete folder "C", you need an exclusive lock on folder "B".

If you are executing multiple sub-processes in parallel that are working on sub-folders, consider the locking behaviors when adding a Delete activity. A Delete activity attempts to get a lock on the parent of the sub-folder that is being deleted.

- If the Delete activity is in a sub-process, you may put the Delete activity in a thread pool of size one to force the deletes in a series and avoid locking conflicts.
- Alternatively, you can put the Delete activity in a parent process and delete the parent folder after all the sub-processes are complete.

Folder or document input variables

When a document is used as input to a capture activity or process. For example, the Partial Completion feature can create a new job for each document created by a Classification activity. Capture activities in the jobs created need an exclusive lock on the document.

If the folder containing the documents is locked and a different user attempts to perform capture activities on the document, the following `error` occurs: The Document cannot be locked since one of its parent folders is already locked. If you need technical assistance, reference action: DocumentReviewControlLoadBatchAction.

Alternatively, if the document is locked and a different user attempt to perform capture activities on the folder, the following `error` occurs: The Folder cannot be locked since its hierarchy contains already locked objects. If you need technical assistance, reference action: ValidationControlLoadbatchAction.

Similar locking problems can happen if you use a looping node or any process design that acts on documents directly.

To avoid locking collisions, be sure your process design has logic to avoid conflicts. One strategy to accomplish this is to use preconditions to force process execution to wait until potentially conflicting jobs or activities are completed.

Web Capture control

When building an ad hoc capture form using the Web Capture control, take care to manage locking behavior. The Web capture control locks the folder used when a scan or upload is initiated. The lock is released when the user clicks the Save button on the toolbar.

Alternatively, if you choose not to display the Save button for users, calling the Save action on the Web capture control will also unlock the folder. This should be done prior to allowing the user of the form to interact with folder content outside the Web capture control and prior to any subsequent capture activities.

Chapter 13

Handle document retention

This section describes the best practices for document retention.

Tables affected by document retention

The following tables in the TotalAgility_Reporting database are affected by document retention.

- batch_accum_fact
- batch_class_dim
- batch_dim
- batch_field_fact
- batch_sess_snapshot_fact
- doc_accum_fact
- doc_class_dim
- doc_dim
- doc_export_fact
- doc_sess_snapshot_fact
- event_data_dim
- field_accum_fact
- field_changes_fact
- field_column_dim
- field_dim
- group_value_dim
- machine_dim
- mod_dim
- object_audit_fact
- page_dim
- path_dim
- product_dim
- reject_note_dim
- station_dim
- ta_categories_dim
- ta_classif_group_dim
- tsf_class_dim
- user_dim

The main table, [doc_dim], contains information about the document. Once the document is considered to be purged, all the associated data is deleted, except when the data is used by users, machines, and other such documents.

In the TotalAgility_Reporting_Staging database, only the [wsa_messages] table is affected by the Unprocessed Incomplete Messages retention policy.

Document retention policy

Frequency and conditions for retention start

Reporting Service executes document retention purge once a day. To determine whether it has performed the retention purge or not, it uses the [retention_stamp] table where it adds the recent DateTime stamp with the current date of purge and 00:00:00 of time. When each ETL cycle starts, it checks if the current time is greater than the last time in [retention_stamp] by a day. If yes, it starts the purge.

Conditions for document purge

Document and associated data are deleted based on one of the following conditions:

- Document was explicitly deleted from the TotalAgility system ([doc_dim].[is_deleted] column)
- Document was deleted via merge of multiple documents in the TotalAgility system ([doc_dim].[is_deleted] column)
- Document was deleted by the TotalAgility documents database retention ([doc_dim].[is_deleted] column)
- Document was marked as completed ([doc_dim].[is_processing_completed] column)

The deletion is also based on either of the following conditions:

- Document falls into the range of data that Tungsten Analytics for TotalAgility has marked as being processed ([etl_job] table, [etl_status] column being equal to -1).
- The document's last processing time ([dt_last_proc_datetime] column) is older than the number of days configured by the Document retention policy.

Order and retries

The Document retention policy is executed before all ETL actions and may take some time. Until the purge is complete, ETL does not proceed. We recommend that you configure running the Reporting service outside of the peak TotalAgility load hours so that the first run of document purge does not overload the system.

If document retention fails due to any issue, a warning is written, and Reporting proceeds to the ETL cycle. The next retry is done the next day. All the previously undeleted data is deleted the next day if there is no other issue.

Default and recommended retention policy values

The default number of days after which documents are deleted from the Reporting database is 3650 days (10 years). We recommend you revise this value and set it to a lower value such as 30 days.

If there is a considerable load of data, you may want to lower that value to the number of days when you are sure the data is loaded but is not needed.

The most used tables include:

- [doc_sess_snapshot_fact]

- [field_accum_fact]
- [object_audit_fact]
- [field_changes_fact]

It might take some time for the Reporting Service to process incoming messages from the [wsa_messages] table, so, data in those tables might appear with some delay. Use the latest data from [dt_end_datetime]/[dt_last_sess]/[dt_action_time] columns to determine what was the latest processed document time. The retention period for documents should not be lower than this number of days.

If the retention value is too high, the mentioned tables become overloaded and may cause lower processing time for each ETL cycle with the same amount of incoming data or cause execution timeouts.

Unprocessed Incomplete Messages retention policy

During the activity execution in TotalAgility, there could be a document split, document deletion, or other such events. Those events are reported as separate wsa_messages entries. Upon completion of the activity, such a list of wsa_messages of events with activity data related to wsa_message forms a complete set of data that ETL is ready to process.

In case of abnormal activity termination, network issues, or other issues, TotalAgility tries to delete the incomplete messages. Still, sometimes it fails to do so, for example, because of a persisting network issue. In this case, a list of incomplete messages remains in the wsa_messages table. They might become complete after TotalAgility retries sending the reporting data; otherwise, they can stay for some time.

The Unprocessed Incomplete Messages retention policy value determines the time those orphans remain in wsa_messages.

Small numbers of orphans mostly do not affect the performance, but this may change if their number is too big. We recommend you leave the default value of 30 days for this retention policy.

Chapter 14

Image quality in the Capture Client image viewer

TotalAgility uses each browser's built-in image scaling algorithms to perform image scaling when displaying images in a non-native resolution in the image viewer. As such, the quality of the image displayed to the user may differ from browser to browser. The effect of this scaling on image quality is outside of TotalAgility's control since TotalAgility relies specifically on each browser to perform this task.

The Chrome browser handles image downscaling better, therefore, if the image display quality is important to your work process, use the Chrome browser.

Chapter 15

Online learning

Online Learning is a method of using unsuccessful classification or extraction results to improve documents processed in the future. If a document is not successfully classified and then correctly classified during production, that document can then be trained so that subsequent similar documents are successfully classified. Corrected extraction results can benefit a project similarly.

Online Learning System task

Online Learning is executed by the TotalAgility system task named Online Learning. This task picks up all documents the users or the system flag for online learning. The system task compiles a new, amended version of a dynamic knowledge base that includes the new documents. Once the task is complete, the newer and better knowledge base is available to the Transformation Server for use in the next job it processes.

Every time the system task runs, it creates a new knowledge base file. This file is not large, but older files are not quickly deleted because there could still be jobs that require them.

Consider the following two best practices:

- Configure the Online Learning system task to run not more than once per hour. Even once per day is usually sufficient, especially in invoice scenarios. Do not have that task run every minute, because it will just create new knowledge base files but the effect of them is only noticeable when a new document of the type that was just learned is processed. For invoices, this does not usually happen before the next day, as a new vendor would not likely send two invoices on the same day. Even if it happens, it is not too much burden for the operator to key those two invoices manually.
- The knowledge base files are stored where TotalAgility is installed which is typically on the C drive. Provide enough space on the C drive if you decide to run the Online Learning System Task more than once a day.

Classification Online Learning

For projects with a Classification Group enabled, Classification Online Learning is configured in the "Advanced Online Learning Options" window in the Transformation Designer. If your Classification Group also has document separation enabled, classification online learning becomes Separation & Classification online learning All labels are updated to include separation online learning.

If your project does not have a Classification Group, it is not possible to configure Classification Online Learning or Separation and Classification Online Learning.

For more information on Classification Online Learning or Separation & Classification Online Learning, see *Online Learning* in the *Transformation Designer Help*.

Documents that are modified during production are collected automatically. However, how the collected documents are used depends on how the project is configured. Either the documents are used immediately in Dynamic Classifiers, or they are collected until they can be imported.

The following online learning for classification or separation and classification process occurs during production:

1. An unknown document is scanned into the documents database.
2. Transformation Server gets the document from the documents database and then performs classification and separation on the document before returning it to the database.
3. An operator performs the document review activity and if the operator determines that the document does not match its assigned document type, they reclassify the document. If the document is incorrectly separated, they correct the separation.

   Upon completion of the activity, if the document was reclassified or separated, the Thin Client returns the updated document to the document database and copies the document to the Online Learning Folder holding area.
4. At scheduled times, the Transformation Server takes all the documents that have accumulated in the Online Learning Folder and trains them. The classification and separation results of the documents are trained, and subsequent processing is improved.

   The training documents are copied to another holding area called New Samples. These documents can be imported by an administrator into a project using Transformation Designer. These imported documents are used to train the project and improve its performance.
5. The next time that the Transformation Server performs classification or separation, the new information in the Dynamic Classifiers is used to produce better results for the operators. The performance of the project improves over time.

Importing training documents into the project for training is done at certain *intervals*. If your project is configured to use Dynamic Classifiers, the accumulated documents, yet to be trained can be used to aid classification and separation during production. If you do not use Dynamic Classifiers, training documents are still collected, but they cannot be used to improve classification or separation results until after they are imported into Transformation Designer and the project is retrained.

It is recommended to:

- use dynamic classifiers to get the most from collected classification and separation training documents.
- import the training documents and train your project for Separation and Classification or Classification at specific intervals.

Extraction Online Learning

Extraction Online Learning is a feature that is configured in the Transformation Designer to learn from documents that were not extracted successfully.

Extraction Online Learning is available only for trainable locators. For more information on Extraction Online Learning, see *Online Learning* and for more information on trainable locator, see *Extraction* in the*Transformation Designer Help*.

The following online learning for extraction process occurs during production:

1. An invoice is scanned into the system and is stored in the documents database.
2. The Transformation Server gets the document from the database and then performs extraction. After extraction, the document and its extraction results are returned to the database.
3. The validation operator takes the validation activity (Thin Client - Document Services) that opens the updated document from the documents database. If any of the extraction results were unsuccessful, the operator specifies the correct information.

   Upon completion of the activity, the validated document is returned to the database with the correct training information. The Thin Client also copies the modified documents to a special holding area in the documents database called the Online Learning Folder.
4. At scheduled times, the Transformation Server will take all the documents stored in the Online Learning Folder and train the project dynamically. Any information that the validation operator validated will be processed and saved in the Dynamic Specific Knowledge Base.

   These documents are also copied to another holding area called New Samples where an administrator can import them into Transformation Designer to help the project perform better.
5. The next time that the Transformation Server performs extraction, the new information in the Dynamic Specific Knowledge Base is used to aid in extraction.

If the scheduled task runs after the next scan activity is run, the Dynamic Specific Knowledge Base is not updated, and the new activity does not benefit from any accumulated documents since the last scheduled task.

ⓘ It is recommended to:

- Use dynamic knowledge bases during extraction
- Enable Automatic training after validation
- Ensure that each trainable field is configured as followed:
  - Show in training dialog
  - Monitor for automatic learning
- Use only those training documents that are clear, have no marks, and that contain all trainable fields for a class
- Import training documents and train your project for Extraction at regular intervals

Intervals for importing training documents

When a project is first put into production, its classification and extraction results are not ideal. Most training documents are accumulated when a project is first placed into production but decrease over time.

As extraction training documents are accumulated, they sit in the Dynamic Extraction Knowledge Base but are only accessed if other extraction methods fail. This means two things:

1. Documents in this knowledge base are not used until all other options fail. This may mean that the extraction performance can suffer.
2. As the Dynamic Extraction Knowledge Base increases in size, the performance of the project may suffer.

To ensure that Dynamic Extraction Knowledge Base does not get too large and that you move your extraction training documents into your Transformation Designer Extraction Set on a regular basis, import your new extraction samples, resolve conflicts, and train your project regularly.

As classification training documents are accumulated, they are used by the Dynamic Classifiers, similarly to extraction, as the training documents increase in number, the performance of the Transformation Server may suffer. It is recommended to import both the classification training documents and the extraction training documents into the Transformation Designer and then train your project for both classification and extraction at regular intervals.

> ℹ You can use the Transformation Designer at any time to see now many documents have been collected and copied into the New Samples set.

Consider the following examples for intervals.

- After one week
- After two weeks
- After three weeks
- After four weeks
- After two months
- After three months
- After six months
- After one year

Set or modify the intervals depending on the volume of documents that you process and the number of trainings that are accumulated.

After one year, your project should be successfully processing documents without problems. The training documents are only collected when a new vendor or form is encountered. Continue to monitor your project and import the document and re-train your project every six months or so.

Chapter 16

Automated export and import of packages

Use the following SDK APIs for the automated **export** of packages:

1. `PackageService – ExportPackageToBytes()`: This API exports the package to a byte array passed from the TotalAgility server to the caller.

   ⓘ This is the recommended approach.

2. `PackageService – ExportPackageToFile()`: This API exports the package to a file that must be accessible from the TotalAgility server.

Use the following SDK APIs for the automated **import** of packages:

1. `PackageService – ExportPackageFromBytes()`: This API imports the package from a byte array passed from the caller to the TotalAgility server.

   ⓘ This is the recommended approach.

2. `PackageService – ExportPackageToFile()`: This API imports the package from a file that must be accessible to the TotalAgility server.

Chapter 17

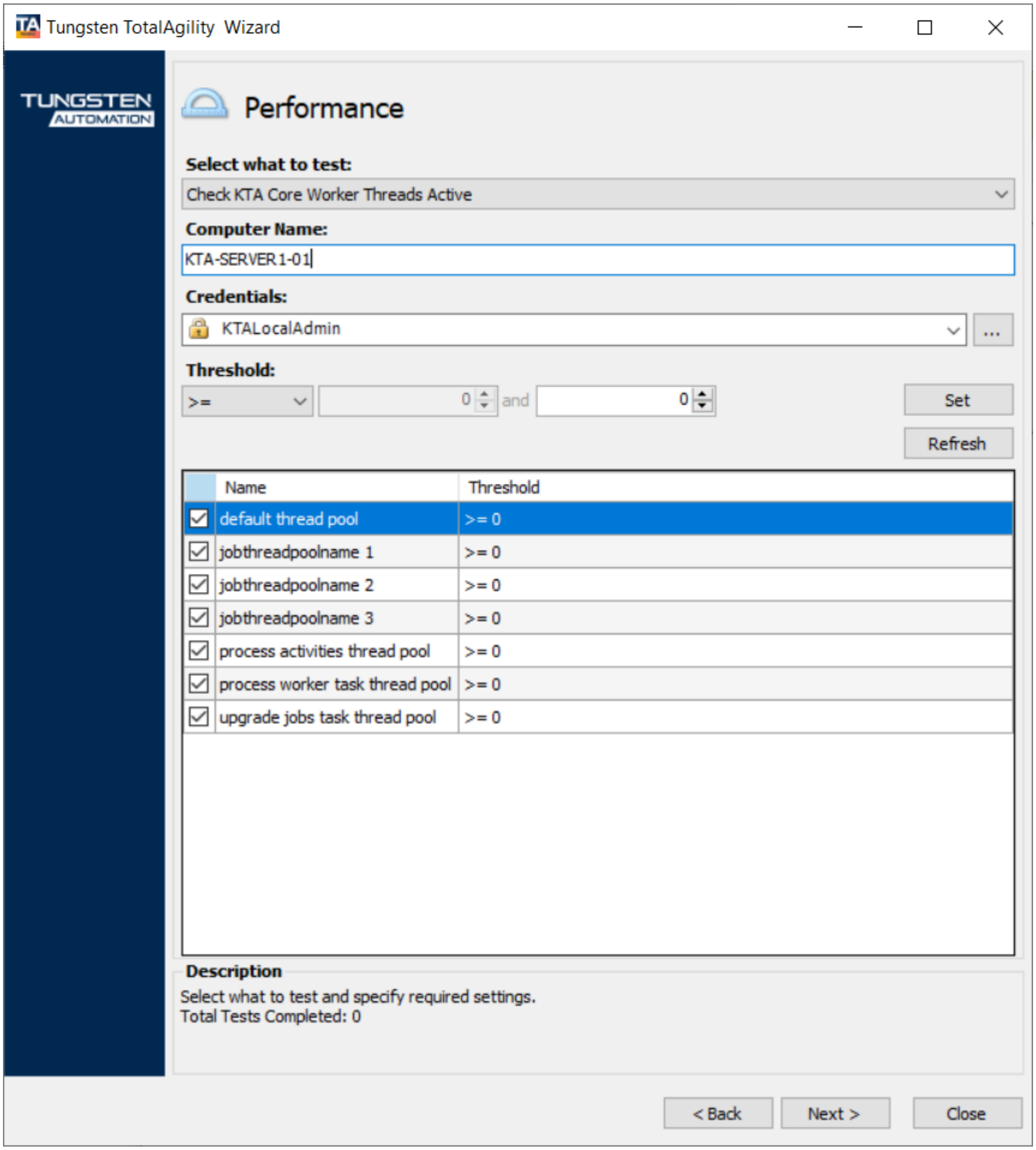
Monitor TotalAgility operating performance

TotalAgility includes Windows Performance Counters to measure key indicators of several critical services. Refer to the "Performance Counter usage" appendix in *Tungsten TotalAgility Administrator's Guide* for more information.

You can use Tungsten Monitor to track these counters and send alerts when the values are outside of defined ranges.

To enable monitoring of the TotalAgility Performance Counters, perform the following steps.

1. Make sure the WMI Performance Adapter Windows service is running on the target TotalAgility machine.
2. Verify WMI connectivity from the Tungsten Monitor server to the TotalAgility server. Use the "Microsoft wbemtest" utility to verify remote WMI access. Refer to the Microsoft website for more information on wbemtest.
3. On the Tungsten Monitor Server, start the **Tungsten Monitor Admin Console** and select **File > New Monitor**. Enter the monitor name, then select the **Tungsten TotalAgility Wizard**, and next the **Performance component** to create TotalAgility Performance Counter tests for the following Performance Counters:
   - Check KTA Core Worker Worker Tasks Taken
   - Check KTA Core Worker Threads Active
   - Check KTA Core Worker Items in Thread Pool
   - Check KTA Core Worker System Tasks Taken
   - Check KTA Core Worker Locked Activities
   - Check KTA Search and Matching Server
   - Check KTA Export Service



Performance

Select what to test: Check KTA Core Worker Threads Active

Computer Name: KTA-SERVER1-01

Credentials: KTALocalAdmin

Threshold: >= 0 and 0

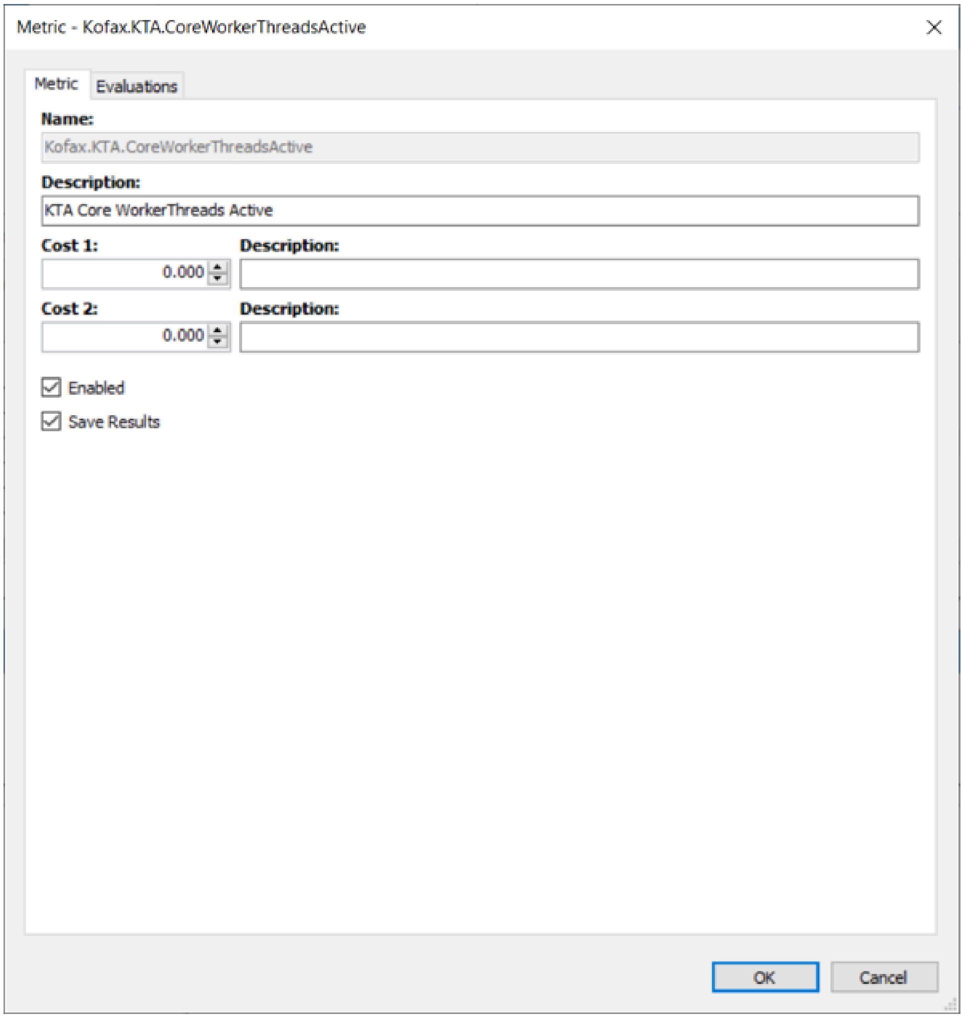
	Name	Threshold
☑	default thread pool	>= 0
☑	jobthreadpoolname 1	>= 0
☑	jobthreadpoolname 2	>= 0
☑	jobthreadpoolname 3	>= 0
☑	process activities thread pool	>= 0
☑	process worker task thread pool	>= 0
☑	upgrade jobs task thread pool	>= 0

Description

Select what to test and specify required settings.
Total Tests Completed: 0

4. Create the TotalAgility test for the desired performance counter and save the results in a metric for a Tungsten Monitor Dashboard display.

   a. Enable the desired TotalAgility performance metric and set desired metric evaluation thresholds.



Metric - Kofax.KTA.CoreWorkerThreadsActive

Name: Kofax.KTA.CoreWorkerThreadsActive

Description: KTA Core WorkerThreads Active

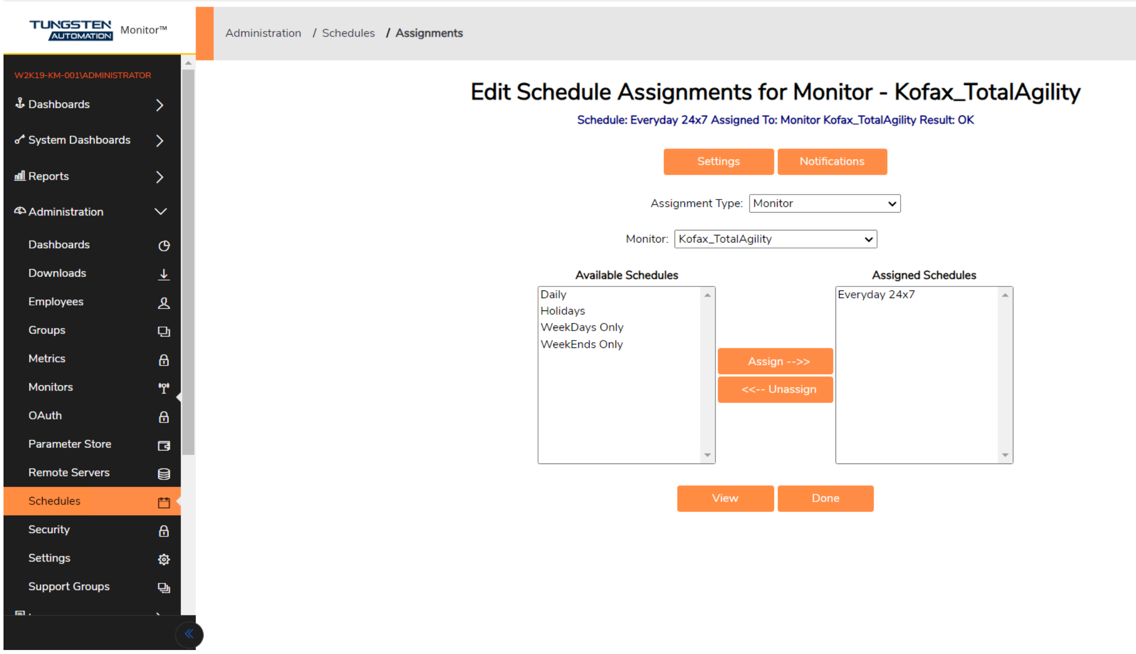
Cost 1: 0.000 **Description:**

Cost 2: 0.000 **Description:**

☑ Enabled

☑ Save Results

b. Assign a schedule and enable notification to the new TotalAgility performance monitor.



Edit Schedule Assignments for Monitor - Kofax_TotalAgility

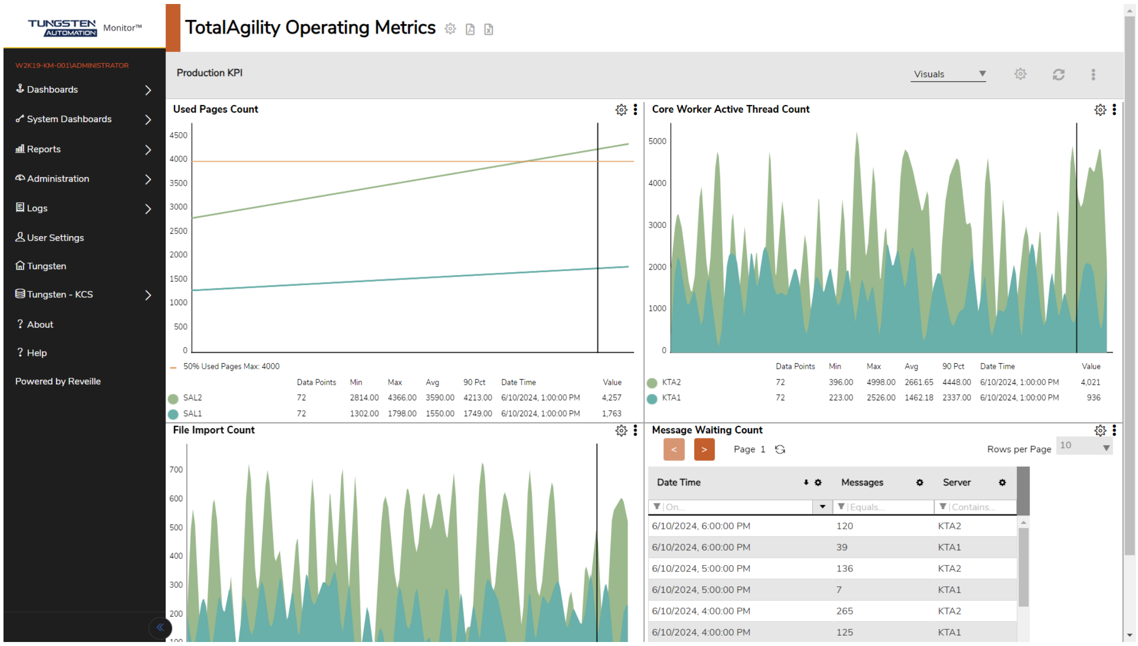
Schedule: Everyday 24x7 Assigned To: Monitor Kofax_TotalAgility Result: OK

Assignment Type: Monitor

Monitor: Kofax_TotalAgility

Available Schedules	Assigned Schedules
Daily Holidays WeekDays Only WeekEnds Only	Everyday 24x7

c. Create a Tungsten Monitor dashboard to display the TotalAgility performance metrics and other desired KPI's.



TotalAgility Operating Metrics

Used Pages Count

50% Used Pages Max: 4000

	Data Points	Min	Max	Avg	90 Pct	Date Time	Value
SAL2	72	2814.00	4366.00	3590.00	4213.00	6/10/2024, 1:00:00 PM	4,257
SAL1	72	1302.00	1798.00	1550.00	1749.00	6/10/2024, 1:00:00 PM	1,763

Core Worker Active Thread Count

	Data Points	Min	Max	Avg	90 Pct	Date Time	Value
KTA2	72	396.00	4998.00	2661.65	4448.00	6/10/2024, 1:00:00 PM	4,021
KTA1	72	223.00	2526.00	1462.18	2337.00	6/10/2024, 1:00:00 PM	936

File Import Count

Message Waiting Count

Page 1 Rows per Page 10

Date Time	Messages	Server
6/10/2024, 6:00:00 PM	120	KTA2
6/10/2024, 6:00:00 PM	39	KTA1
6/10/2024, 5:00:00 PM	136	KTA2
6/10/2024, 5:00:00 PM	7	KTA1
6/10/2024, 4:00:00 PM	265	KTA2
6/10/2024, 4:00:00 PM	125	KTA1

ⓘ For a complete description of Tungsten Monitor out-of-the-box support for TotalAgility, refer to the "Use the Tungsten Monitor Wizards" document included with the Tungsten Monitor documentation.

Chapter 18

Monitor TotalAgility linked servers

The configuration of TotalAgility linked servers defines a two-way link between two separate TotalAgility installations for moving documents between the systems. Refer to "Linked Servers" in the *Tungsten TotalAgility Designer Help*.

The Distributed Server feature of Tungsten Monitor enables the Tungsten Monitor Server to operate at multiple TotalAgility linked server locations proactively, running the same or different sets of TotalAgility Monitors at each location. Tungsten Monitor Remote Servers run local TotalAgility Monitors and send monitoring results over encrypted Web Services connections. Optionally, local alerts can also be sent to one or more Tungsten Monitor master servers.

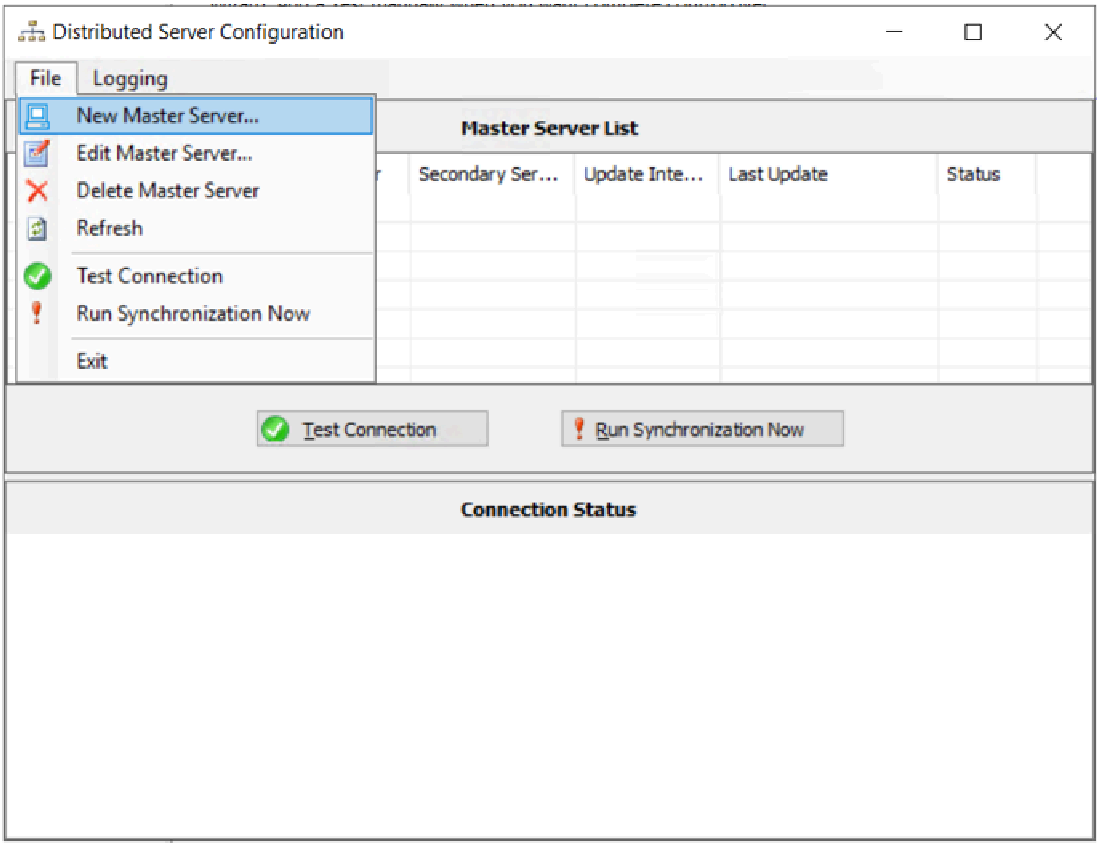
You can view the status and reports of Tungsten TotalAgility Remote Monitors as if the Monitors are located and running at the Tungsten Monitor master server location.

To enable the Distributed Server feature of Tungsten Monitor for TotalAgility linked servers, perform the following steps.

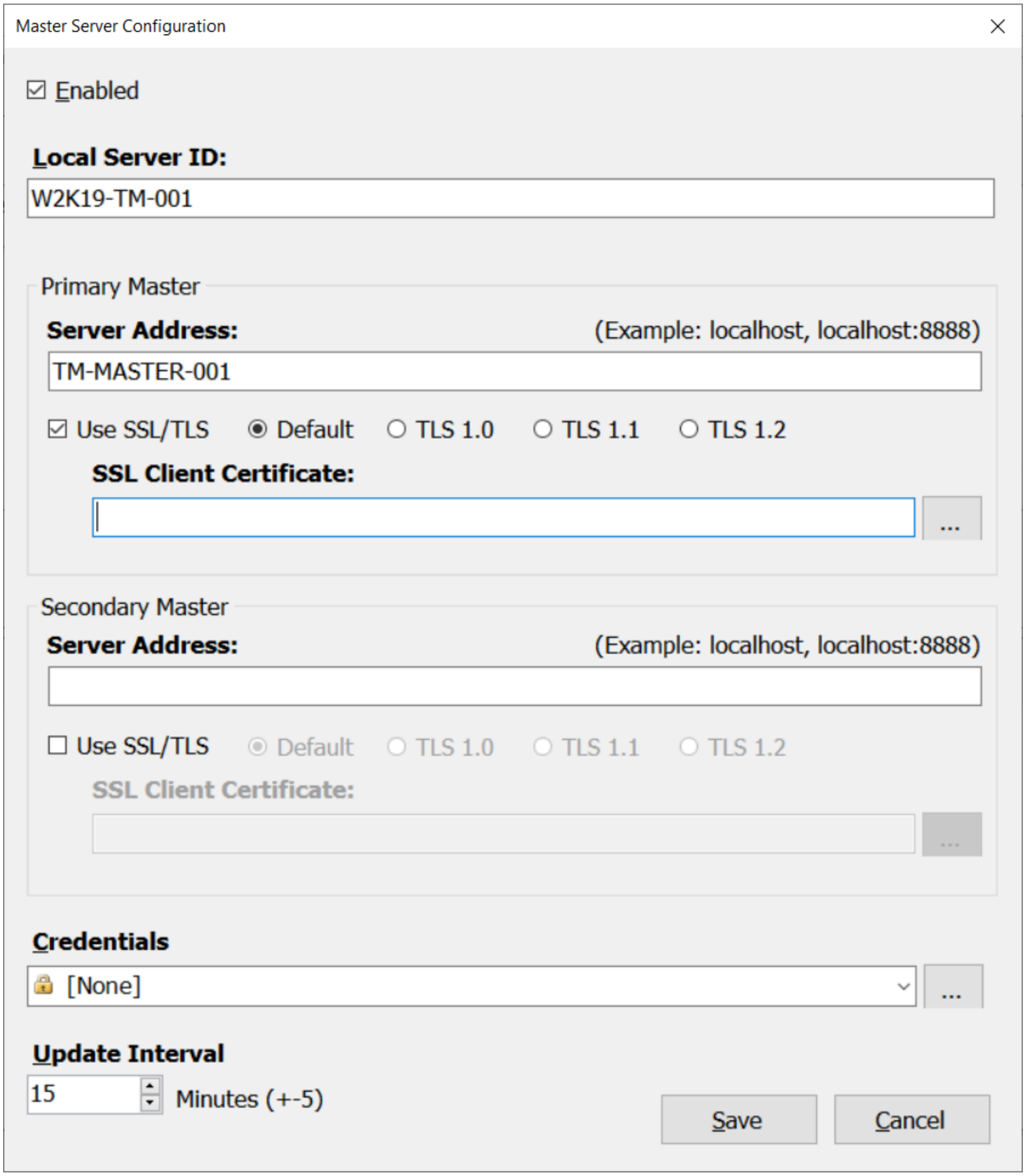
1. Install Tungsten Monitor at each TotalAgility linked server location. Refer to the *Tungsten Monitor Installation and Setup Guide*.
2. Create TotalAgility Monitors at each TotalAgility linked server location using the TotalAgility Wizard.

   Refer to the “Using the Tungsten TotalAgility Wizard” section in the *Using Tungsten Monitor Wizards Guide*.
3. Create a Tungsten Monitor distributed server connection by configuring the Tungsten Monitor distributed server at the Tungsten Monitor Admin Console. The Tungsten Monitor distributed server connection uses the web services with optional SSL encryption to communicate between distributed Tungsten Monitor servers.

   a. At the Tungsten Monitor remote server, start the **Tungsten Monitor Admin Console**, and select **File > Distributed Server Configuration**.



b. Select **File > New Master Server**.



Master Server Configuration

☑ Enabled

Local Server ID: W2K19-TM-001

Primary Master

Server Address: (Example: localhost, localhost:8888)

TM-MASTER-001

☑ Use SSL/TLS ● Default ○ TLS 1.0 ○ TLS 1.1 ○ TLS 1.2

SSL Client Certificate:

Secondary Master

Server Address: (Example: localhost, localhost:8888)

☐ Use SSL/TLS ● Default ○ TLS 1.0 ○ TLS 1.1 ○ TLS 1.2

SSL Client Certificate:

Credentials

[None]

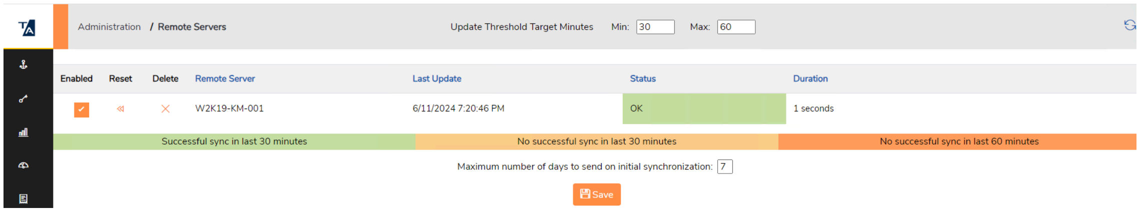
Update Interval

15 Minutes (+-5)

c. Define the Tungsten Monitor master server connection and click **Save**.

d. Save the master server configuration.
The connection is disabled until explicitly enabled at the Tungsten Monitor master server.

e. To enable the Tungsten Monitor remote server connection, perform the following steps:



Update Threshold Target Minutes Min: 30 Max: 60

Enabled	Reset	Delete	Remote Server	Last Update	Status	Duration
☑	«	×	W2K19-KM-001	6/11/2024 7:20:46 PM	OK	1 seconds

🟩 Successful sync in last 30 minutes 🟨 No successful sync in last 30 minutes 🟧 No successful sync in last 60 minutes

Maximum number of days to send on initial synchronization: 7

1. Select Tungsten Monitor User Console **Administration** > **Remote Servers**.
2. Enable the connection.
3. Click **Save**.

f. To verify the connection from the Tungsten Monitor remote server, on the Distributed Server Configuration, click **Test Connection**.