Lexmark™
Enterprise Software

Kofax TotalAgility®
Implementation Best Practice Guide

TABLE OF CONTENTS

1. Overview

1.1. Purpose

This document provides best practice guidelines and recommendations for architecting, provisioning and maintaining a Kofax TotalAgility platform, and designing, implementing, deploying, monitoring and supporting solutions based on the TotalAgility platform and related technologies. In addition to guidelines on the TotalAgility product, advice is also included in respect of the use of Kofax Insight as a component of a TotalAgility platform implementation.

It has been authored by the Centre of Excellence team within Kofax Professional Services Group, with the aim to support consultants and other teams both within the Kofax\Lexmark organization and within the partner and customer community. It aims to help these teams understand implementation complexities and considerations of the components of the TotalAgility platform, better enabling them to support large and complex implementations of the product within customers' environments and to ensure that these solutions are fit-for-purpose, maintainable and supportable.

This document complements other TotalAgility documentation available to licensees of the product. Please use this document in combination with product documentation, such as installation and other guides (See reference section that follows for some of the recommended documentation).

1.2. Intended Audience

Consultants, technology leads, project managers and technical specialists who will be architecting, deploying and supporting the Kofax TotalAgility platform, and who will be developing solutions using the platform and related technologies.

1.3. Revision History

Issue	Date	Author	Comments
1.0	July 2016	Kofax PS COE team	First publication

1.4. References

This document should be used in conjunction with official Product documentation and is intended to complement that documentation.

The documents highlighted in the table below either provide additional information or are sources for information included in this document.

Document Title	Author	Description
Kofax TotalAgility 7.x Installation Guide	Product Team	Provides installation prerequisites, sizing and architecture guidelines for installing the TotalAgility product components and step-by-step installation instructions.
Insight Configuration for High Availability and Disaster Recovery	Product Team	Provides options for installing and configuring Insight Analytics for High Availability and Disaster Recovery.
Kofax TotalAgility Features Guide	Product Team	Provides an overview of the features provided by the TotalAgility platform.
Kofax Search and Matching Server Installation Guide	Product Team	Provides step-by-step instructions for installing the Kofax Search and Matching Server.
Kofax Insight Installation Guide	Product Team	Provides step-by-step instructions for installing and configuring Kofax Insight Analytics.
Kofax Analytics for TotalAgility Administrator's Guide	Product Team	Provides step-by-step instructions for deploying and configuring the KAFTA/KIAFTA solution for Insight Analytics and TotalAgility.

2. Glossary

Acronym	Description
DB	Database
ETL	Extract, transform and load
ICR	Intelligent Character Recognition
IIS	Internet Information Services, a Microsoft software product
KC	Kofax Capture
KAFTA	Kofax Analytics for TotalAgility - Analytics solution for TotalAgility built using Kofax Insight Analytics. Provides a set of out-of-the-box views and dashboards that cover Capture, Extraction and Business Process Intelligence.
KIAFTA	Kofax Insight Analytics for TotalAgility - the same Analytics solution as KAFTA, but licensed to allow also the creation of additional records and metrics based on the TotalAgility datasource.
KTA	Kofax TotalAgility
KT	Kofax Transformation (formerly known as KTM, Kofax Transformation Modules)
KCS	Kofax Communications Server
KIC	Kofax Import Connector (imports emails/folders into Kofax)
KSMS	Kofax Search and Matching Server
OCR	Optical Character Recognition
OMR	Optical Mark Recognition
RTT	Real Time Transformation
SQL	Structured Query Language
SDD	System Detailed Design
SSL	Secure Socket Layer

TD	Transformation Designer
TLS	Transport Layer Security
TS	Transformation Server
Batch	Group of scanned documents to be processed together
Classification	Identification of the document type based on key markers found. This is used downstream during extraction to only extract the required fields based on the document type found.
Extraction	Obtain key data fields from the document in order to perform future business action. This can be a business rule being applied, calculations performed, or pushing the data into a third party application to prevent double-data entry.
Validation	Providing a confidence level or human interaction to the technical process of Classification and Extraction should it be required. If the extraction results meet or exceed the configured confidence levels the document to will onto the downstream process. If these levels are not met then the document will be presented to an end user for visual verification.
VRS	Virtual ReScan
WAF	Web Application Firewall

3. Architecture Guidelines

This section provides guidance and best practice for architecting a TotalAgility platform, with considerations based on the type of customer and solution being deployed.

3.1. Types of Deployments

Kofax TotalAgility supports a wide variety of platform deployments, and the individual components of the solution can be scaled horizontally or vertically, depending on the specific use case for the customer and the target solution.

Some of the factors that influence the complexity of the platform include the customer's internal compliance requirements, their approach to resiliency and redundancy within the platform, use of virtualisation technologies and security requirements.

At a very high-level, the types of deployments can be categorised as:
- Low complexity or low volume solutions;
- Medium to high complexity or high volume solutions.

The approach and considerations for these different types of deployment models are described below.

3.1.1. Low Complexity

A solution that is low complexity could be characterised as a point solution or a simple use-case, a platform operating in an environment without strict regulatory compliance or a non-critical business system.

In these type of cases the solution could be deployed on a small number of servers, with Web/App Servers, Transformation Server and Database Server typically installed separately.

This type of platform could be installed on multiple nodes to provide load-balancing and high-availability but this requirement would need to be driven by the customer, the maturity of the IT organisation, their infrastructure availability and high availability requirements. The usage and business criticality of the solution will also drive help these requirements.

3.1.2. Medium or High Complexity

A medium or high complexity solution could be categorised as a customer facing system requiring platform resiliency, a platform operating in a strict regulatory environment, a high volume solution or a solution with strict SLAs for processing and penalties for non-compliance.

This would typically be for a large customer, with mature IT processes, requiring multiple nodes for load-balancing and high-availability, separate service accounts to operate each component of the platform, and the possibility of strict requirements around separation and zoning of components and the management of data stored within the platform.

In this scenario there could be requirements to install each component separately, including separation of TotalAgility Web and App tiers if required to separate the user-facing and application facing components, separation of License Server and Reporting Server, and Analytics Web and App tiers. In this model each component can be scaled individually by installing additional dedicated servers, e.g. additional Application or Transformation Servers. A hardware or software load balancer would typically be required for this type of configuration, as well as a non-Production environment that replicates the deployment of the same server roles, but with a single tier deployment for each node.

Depending on the volume of documents being processed for capture and extraction and the number of business processes being performed, it may be difficult to size the platform accurately upfront, and there should be an expectation that the platform will scale over time. Each component of the platform should be deployed in a way that allows for easy scaling of the individual components, such as the separation of the License Servers (with the provision of a primary and backup License Server), separation of Web/App Servers, and Transformation Servers (with consideration for the creation of multiple Transformation Server pools to allow for processing of different types of jobs with differing priorities where required).

It is recommended to engage with a Technical Architect to support the design of this type of platform, with ongoing engagement for management and tuning of the system over time. It is expected that there will be high-level of engagement with the customer's infrastructure experts to design the platform architecture, to ensure that the platform meets the requirements of the customer's internal IT policies, particularly around security zoning, the use of service accounts for installation and operation of the services, use of SSL certificates, deployment of the database server/instances and managing the usage of the underlying hardware resources.

For a high-volume solution, it is recommended to use dedicated hardware resources, at least for the Transformation Server and SQL Server components. The use of virtual infrastructure should be avoided for these components, or at least dedicated infrastructure should be provided where the ratio of physical to virtual CPUs is close to 1:1. Overloading of CPU resources can cause excessive wait times, which can be detrimental to the performance of this type of solution.

It is also recommended to pay close attention to the configuration of the Microsoft components in this type of solution, IIS and SQL Server. Default installations of these components would typically not be tuned for a high-performance environment so careful tuning of these components to make the best use of the underlying infrastructure is recommended. Some areas to look out for include memory usage within IIS, number of concurrent sessions and threads, separation of databases used by the solution, and separation of storage of the underlying database and transaction log files, to maximise IO throughput.

3.2. Deployment Models

Kofax TotalAgility provides three main deployment models for the platform:

- **On-Premise** - a single instance platform of TotalAgility, with a set of dedicated software components and database storage intended for a single customer use;
- **On-Premise Multi-Tenancy** - a single platform, supporting multiple instances of TotalAgility, each with its own database, design-time and runtime components. Multi-

tenancy provides a shared execution model, but provides separation for the transient data and data-at-rest within the platform;

- **Microsoft Azure** - both the single platform and multi-tenant deployments of TotalAgility are available in the Microsoft Azure cloud platform, as a Platform as a Service (PaaS) subscription service. Contact your account representative if Azure deployment is your specific interest.

3.3. **Sizing**

Sizing of a TotalAgility platform should be done in context of the type of installation and solution being deployed, i.e. single tenant, multiple tenants, expected volumes of capture and extraction of documents and the number of business processes being executed, the requirements of the customer in terms of high-availability, the business critical nature of the solution, the regulatory environment that the customer is operating in and any SLAs that the solution needs to meet in terms of volume and throughput.

Each component of the solution can be scaled independently of each other, including:

- **TotalAgility Web Server** - scaled based on concurrent usage of the web applications;
- **TotalAgility Application Server** - scaled based on concurrent usage of the web applications, volume and throughput of automated activities within the cases and business process and integrations with third-party systems;
- **Transformation Server** - scaled based on volume and throughput of documents and pages being classified and extracted, the classification method, the quality and type of images being processed, number of documents being processed in each job, and number of pages in each document, and requirements for on-demand processing of jobs (e.g. ingested real-time from a mobile client);
- **License Server** - typically deployed as a two-node, active/passive configuration;
- **Reporting Server** - typically deployed as two-node, active/passive configuration as per the License Server;
- **Import Message Connector** - can be scaled based on the volume and throughput of email files being ingested into the system. Typically co-located with the Application Server but can be installed stand-alone. For solutions with requirements for enterprise-level, high-volume ingestion from multiple sources, the Import Connector can be swapped out for a dedicated Kofax Communication Server (KCS);
- **Kofax Search and Matching Server** - scaled based on the size of the database/files being ingested for use in Extraction lookup and validation. Can be scaled horizontally using a load-balancer;
- **Kofax Analytics Server** - the Web and App tiers of the Analytics Server can be installed separately, and scaled separately, based on concurrent usage of the web applications, and the volume of data being processed, either as part of a scheduled data load or on-demand. This will create additional load on the database server and should be taken into account when performing sizing of the database components;
- **SQL Server Database** - the database is core to the TotalAgility platform, as all design-time and runtime components of the solution are persisted in the database. Careful attention needs to be taken to size the database server resources and storage requirements, including separation of databases for large-scale implementations and usage of the underlying disk storage and connectivity between the TotalAgility servers and the database server. The SQL Server used by TotalAgility should be installed separately from the other components of the solution, except in very small scale implementations of the platform. The databases used by TotalAgility should be separated from those used by the Analytics Server, as usage by the Analytics

components can create additional load on the database server that can affect the performance and throughput of the other components of the solution.

For medium and large scale implementations of TotalAgility, it is recommended to perform performance testing, tuning and sizing in a Test or Pre-Production environment, based on the processes and documents that will be expected in the Production environment for the solution.

As there are many factors that affect the performance of the system, the platform should be designed in a flexible manner, to allow for scaling the components if the desired performance does not meet a customer's expectation based on the real-life testing of the system. There is a special team within Professional Services group, the Technical Architects (TA), which specialises in supporting sizing, scaling and troubleshooting installations for ensuring best practices beyond the product to the underlying infrastructures including but not limited to the Operating System, database, web server, storage etc.

3.3.1. Transformation Server Sizing

Transformation Server concurrency is based on the number of available processing slots and is sized based on the number of CPU cores assigned to the server (physical or virtual). The number of slots is calculated at 1.25 * the number of cores, e.g. a Transformation Server with 8 CPU cores will allow for 10 processing slots.

Each processing slot equates to a concurrent Transformation Server process, and can be used to size the expected throughput, based on the expected document and page volume, and the average time to process each page.

3.3.2. Database Sizing

This section is intended to provide a high-level overview of sizing the product databases, based on expected jobs being processed, volume of documents and pages and average sizes (include overhead for storage of the OCR and fields in the database), accounting for retention of completed jobs, audit trail, document retention and reporting and analytics.

The sizing of the database should take into account the amount of information being extracted, the level of validation/keying required based on expected accuracy of extraction, transaction logs, backups etc.

In the event that this guidance does not meet customer requirements, it should be augmented with advice from Professional Services in consultation with the Technical Architecture team.

3.4. Tiering

TotalAgility can be deployed across 3 load-balanced tiers, though commonly the Web and App tier are combined.

Typically on a scaled out environment the various services in the Application tier can have dedicated servers (e.g. for Transformation Server or for Import Connector).

Virtualization is the most typical deployment approach for all tiers, though physical Transformation Server(s) or Database Servers are also sometimes used.

3.5. **Scalability and Resiliency**

The TotalAgility Installation Guide provides guidelines on scaling and resiliency specific to each version of the product. The Web and Application Server components of TotalAgility consist of a suite of stateless WCF .NET web services. This allows for many different options to load-balance and failover the solution, such as software and hardware load balancers.

Some components provide native application load-balancing as they work on a pool of requests, such as the Core Worker process on the Application Server and the Transformation Server services.

For medium and large scale platforms, it is recommended to load-balance the Web and Application Server components using a hardware or software load balancer. This applies to both the TotalAgility and Analytics components.

The TotalAgility platform can be cross-site load-balanced across two data centers, with the assumption that the SQL Server infrastructure can be clustered or mirrored in real-time across both data centers, and that the underlying network infrastructure has the bandwidth and latency to support this.

The following diagram shows an example architecture for a highly-available TotalAgility on-premise platform using a hardware load balancer to load-balance and failover requests to the TotalAgility Web and Application Servers.
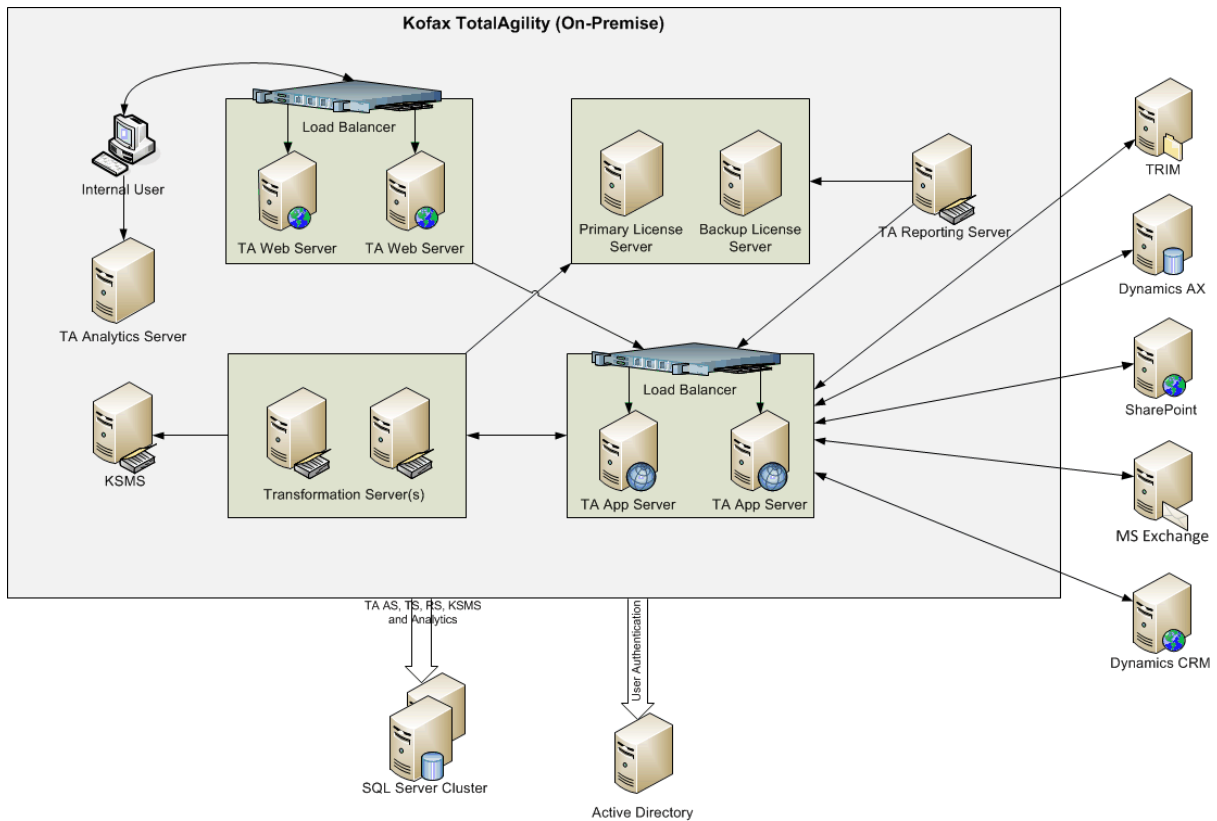
The following table provides an overview of how each component can be scaled, and the minimum number of servers required in a highly-available configuration. Depending on the size and requirements of the customers' environment, some components can be co-located, such as the License Server and Reporting Server, which have a similar scaling and failover profile.

Server Type	Version	Minimum Number of Instances	Description
Kofax License Server	7.2/7.3	2	A primary and backup instance of the Kofax License Server is required to support high-availability. The License Servers run as Active/Passive and the application will automatically failover to the backup instance of the License Server if the primary instance fails.
Kofax TotalAgility Web Server	7.2	2	A minimum of two TotalAgility Web Servers are required to provide high-availability. The Web Servers require a load balancer to manage connections to the servers, as a pool of available servers. The load

			balancer will bypass servers that are unavailable.
Kofax TotalAgility Application Server	7.2	2	A minimum of two TotalAgility Application Servers are required to provide high-availability. The Application Servers require a load balancer to manage connections to the servers, as a pool of available servers. The load balancer will bypass servers in the pool that are unavailable.
Kofax Transformation Server	7.2	2	A minimum of two Transformation Servers are required to provide load-balancing and failover. The servers operate using a request pool and are recommended to run as active/active nodes against the same pool. Additional transformation server pools can be added, and configured to run as single node or multi-node to process high-priority requests. Otherwise all requests will be treated as the same priority and processed in the order that they are received. The platform can be scaled up with additional Transformation Servers instances as required to meet an increased demand and improve throughput. **N.B**. for purposes of Real Time Transformation a Transformation Server should be co-located with the Application Server, so that the synchronous calls RTT uses can perform better than if the Transformation Server were located on a separate physical server.
Kofax Search and Matching Server	6.5	2	KSMS provides its own proprietary clustering mechanism. The primary host connects to the source databases for auto-updates and replicates to all other 'slave' instances. The load balancer routes search requests across the KSMS nodes and

|			
			keeps the master node aware of slave node status (for data replication). Additional hosts are configured in Transformation Designer when adding a connection to a Remote Fuzzy Database, and can be selected for use in Extraction, Validation or both.
Kofax Reporting Server	7.2	1	A single active Reporting Server is required, however additional Reporting Server instances are installed on the TotalAgility Application Servers, and can be started as additional nodes if the primary node fails. Alternately an additional standalone Reporting Server can be installed, and run as a passive node for failover.
Kofax Analytics Server	Insight 5.1 KAFTA 1.2	2	At least two Analytics Servers will be required within the platform. The web applications hosted on the Analytics Server can be load balanced using a hardware or software load balancer. The Insight Scheduler and Bridge services that run on the Analytics data to process the source data sources and prepare the information for the reports can only run as active on a single node.

3.6. **High Availability and Disaster Recovery**

A customer's requirements for high availability and disaster recovery should be understood at the outset of a project.

Where high availability is a requirement, then the physical architecture should seek to avoid single points of failure across all solution components.

3.6.1. **KTA Core Components**

The diagram below depicts the core KTA components and indicates for each component how that component can be deployed in a resilient manner or to provide both resiliency and scalability.
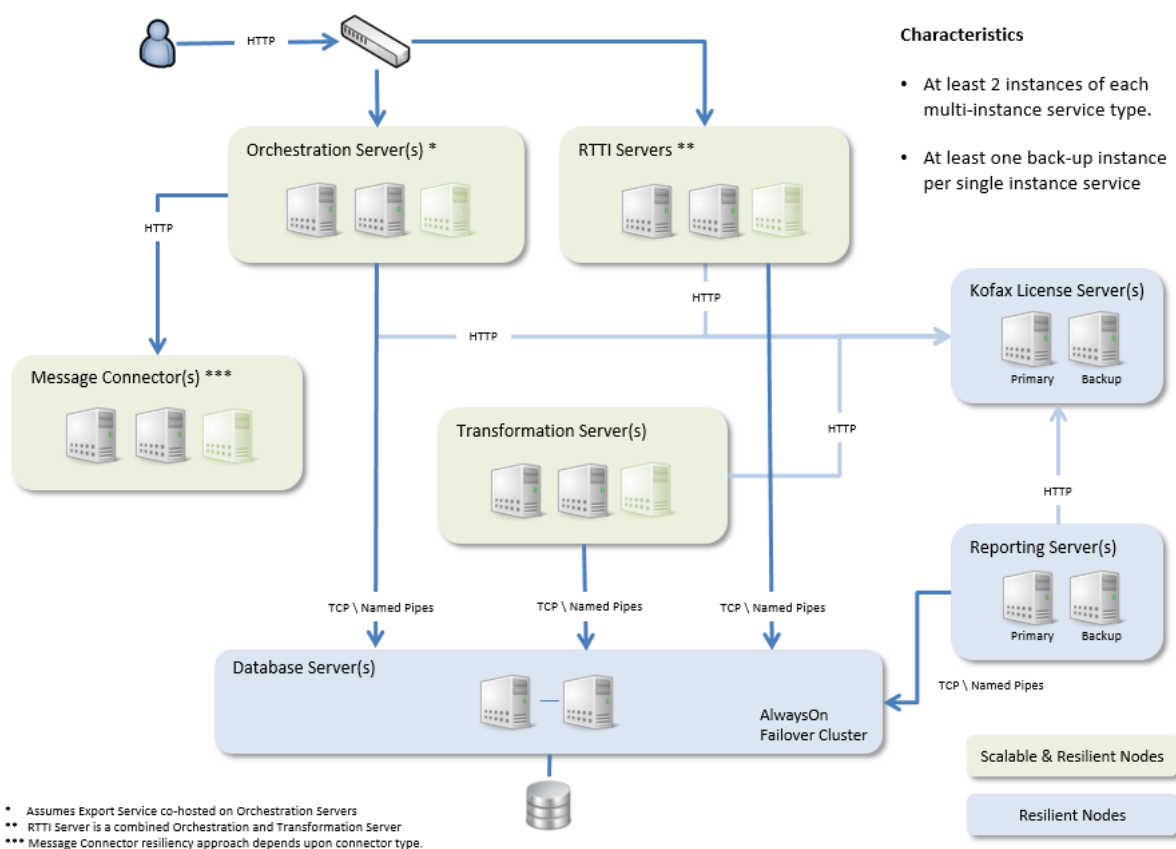
Note: It is assumed that:

- *the underlying database storage is suitably resilient.*
- *the network infrastructure is suitably resilient, including load balancers.*

3.6.2. **KSMS**

The diagram below depicts the KSMS components (and associated KTA components) and indicates for each component how that component can be deployed in a resilient manner or to provide both resiliency and scalability.
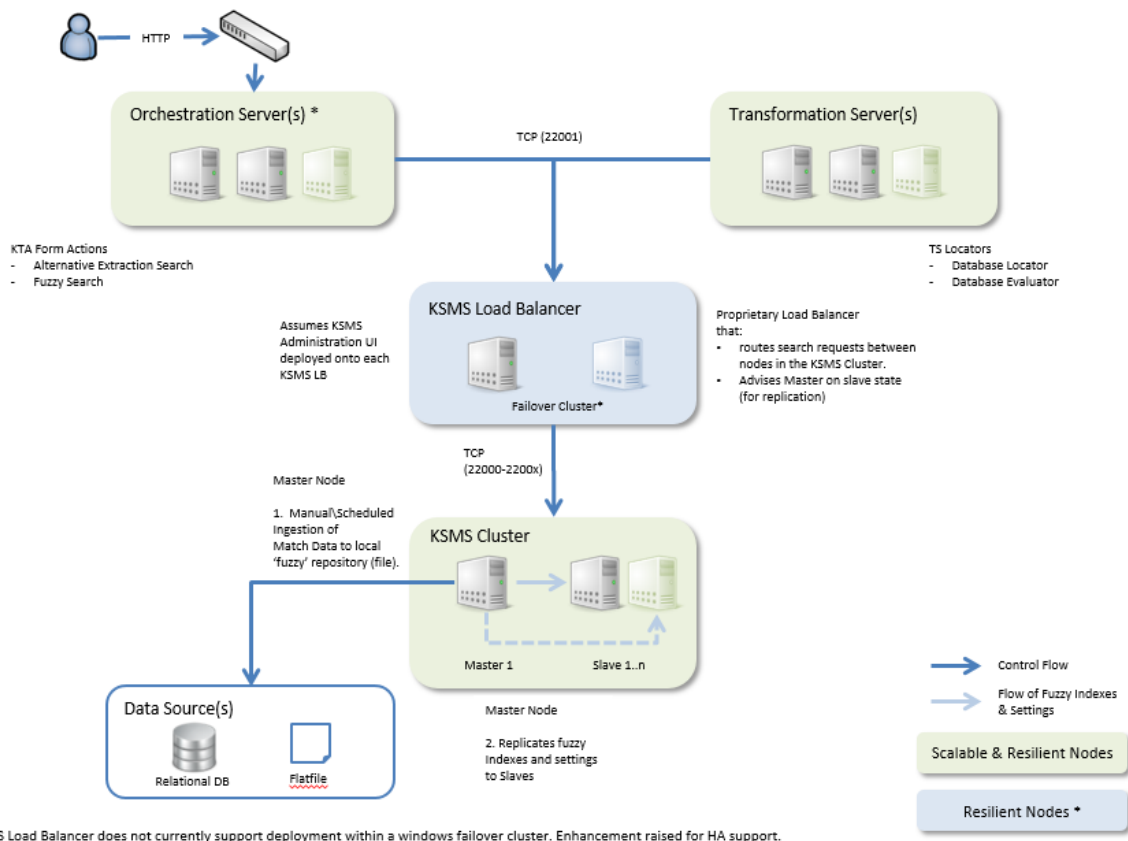
HTTP

Orchestration Server(s) *

TCP (22001)

Transformation Server(s)

KTA Form Actions
- Alternative Extraction Search
- Fuzzy Search

TS Locators
- Database Locator
- Database Evaluator

Assumes KSMS
Administration UI
deployed onto each
KSMS LB

KSMS Load Balancer

Failover Cluster*

Proprietary Load Balancer
that:
- routes search requests between
  nodes in the KSMS Cluster.
- Advises Master on slave state
  (for replication)

TCP
(22000-2200x)

Master Node

1. Manual\Scheduled
Ingestion of
Match Data to local
'fuzzy' repository (file).

KSMS Cluster

Master 1   Slave 1..n

Control Flow

Flow of Fuzzy Indexes
& Settings

Data Source(s)

Relational DB   Flatfile

Master Node

2. Replicates fuzzy
Indexes and settings
to Slaves

Scalable & Resilient Nodes

Resilient Nodes *

* KSMS Load Balancer does not currently support deployment within a windows failover cluster. Enhancement raised for HA support.

Note: It is assumed that:

- *any data source storage (database or file system) is resilient.*
- *the network infrastructure is suitably resilient, including load balancers.*

3.6.3. Insight
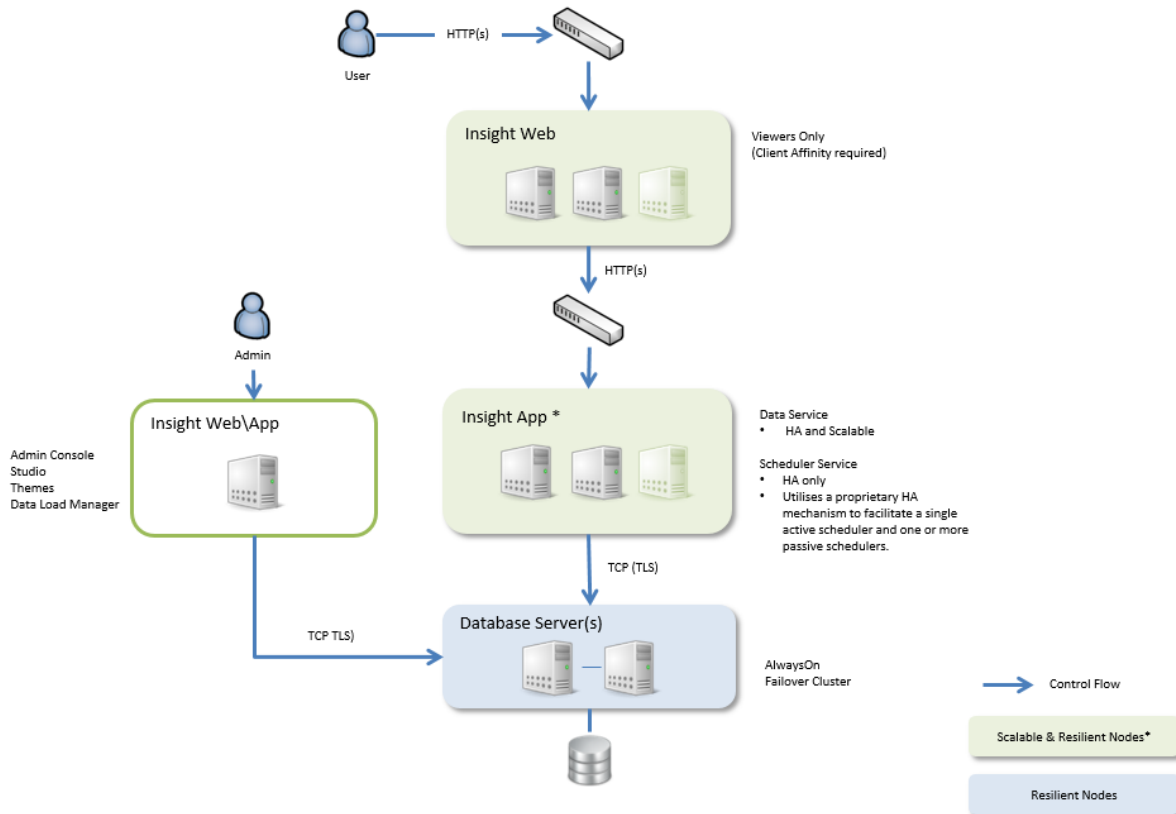
Note: It is assumed that:

- *the underlying database storage is suitably resilient, including any data sources.*
- *the network infrastructure is suitably resilient, including load balancers.*

3.6.4. Disaster Recovery

To meet disaster recovery (DR) requirements:

- The deployment should provision a replica installation at the DR site:
  - The DR site may represent an exact replica of live production where the hosted application is business critical, or a cut down deployment to facilitate more relaxed SLAs or lower volume priority processing only during a DR scenario.
- The underlying product databases must be backed-up and\or replicated to the DR site with sufficient frequency to meet Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO).

Note:

- The responsibility to select the mechanism and provision the failover of the underlying storage (e.g. SAN Replication or SQL Server AlwaysOn Availability Groups) is the responsibility of the customer.
- In some cases the platform can be deployed as active/active across multiple sites. However this requires the database infrastructure to be mirrored in real-time across all sites and for there to be sufficient network bandwidth and latency to support the underlying network traffic between the servers. In this case the secondary sites should be able to process sufficient load and concurrency to support all sites, in the event DR is required.

3.7. Platform Security Requirements

It is important to capture the security requirements at the outset of a project to ensure that these requirements are met by the deployment architecture, configuration and solution implementation.

The customer will ultimately determine the security requirements and approve the approach taken to meet those requirements based upon a combination of factors including:

- Customer standards
- Industry standards
- General best practice and guidelines
- Compliance requirements (e.g. PCI DSS, HIPPA)
- Security vendor (hardware and software) recommendations
- Budget

User management and authentication within the application are discussed in the section titled Application Security.

3.8. Intranet Deployment

The diagram below presents an overview of the (default) ports and protocols involved in the configuration of a secure (intranet) deployment of the core KTA components.
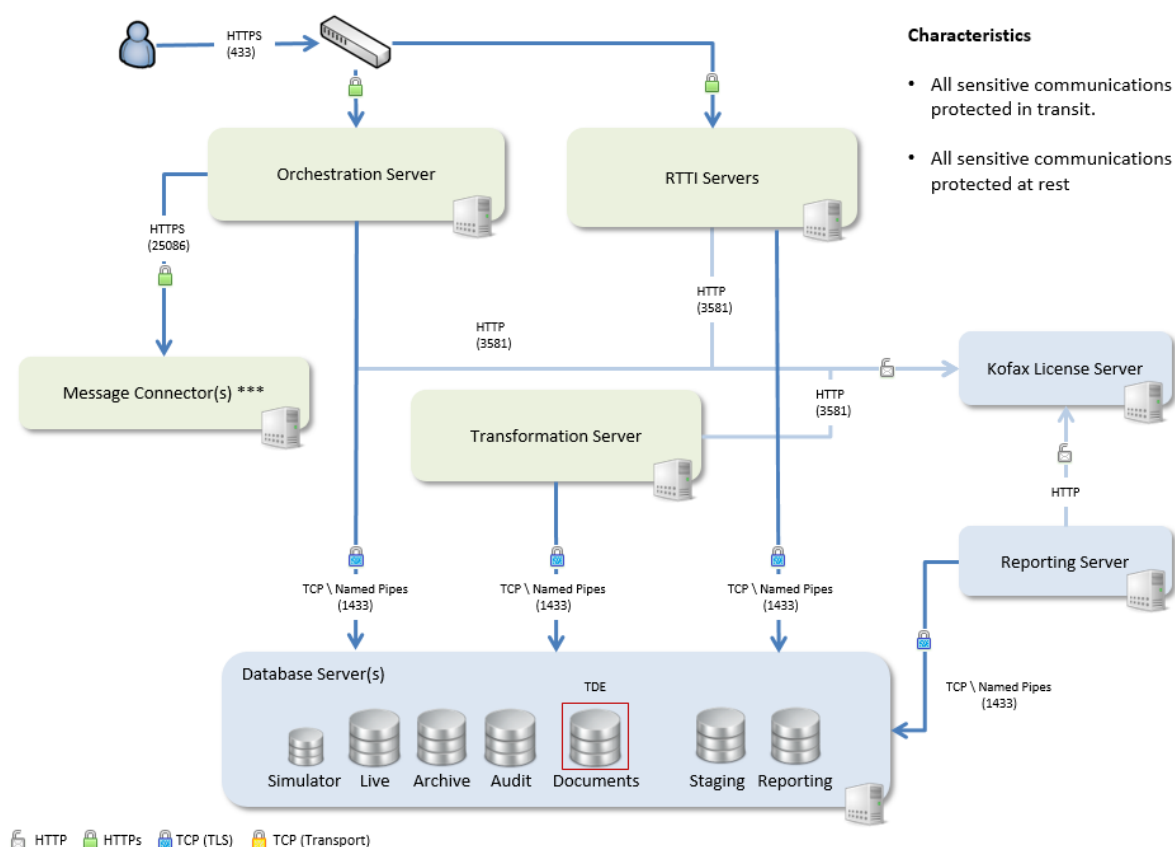
3.8.1.1. Data In-Transit

- Typically SSL/TLS is used to protect data in transit between client browser and web server (and between Web and App server in situations where they are split and communicate using the HTTP binding).
- SSL can also be used to encrypt the data being sent between the TotalAgility servers and the database server where required, although these servers are normally located in the same network zone within the customer's data center.

3.8.1.2. Data At-Rest

- Data residing within SQL Server, can be encrypted using SQL Server Transparent Data Encryption (TDE).
- Data residing on the filesystem can be encrypted using hardware or software encryption where required, or the customer may implement 'compensating controls' which enables the storage location to meet security requirements even if encryption is not in use.

3.9. Internet Facing Deployment

In an internet facing scenario, externally facing components are typically deployed within a perimeter network (i.e. DMZ) protected by a combination of hardware\software firewalls and reverse proxies.

With this deployment type, it may be required that KTA is deployed in a tiered fashion with the web components deployed into the perimeter network and the application server components deployed into the internal network.

This configuration seeks to restrict the open ports and protocols required between the DMZ and internal network, and in particular to eliminate the need for a component in the DMZ to have direct access to backend databases and\or other systems.

3.9.1. Reference Topologies - TotalAgility

The following subsections depict a number of reference architectures for an internet facing deployment of KTA.

Please note:

- The reference topologies depicted are intended to serve as a guide only, to inform design discussions surrounding the deployment options for KTA components.
- It is the responsibility of the customer to provision infrastructure to facilitate the security requirements.
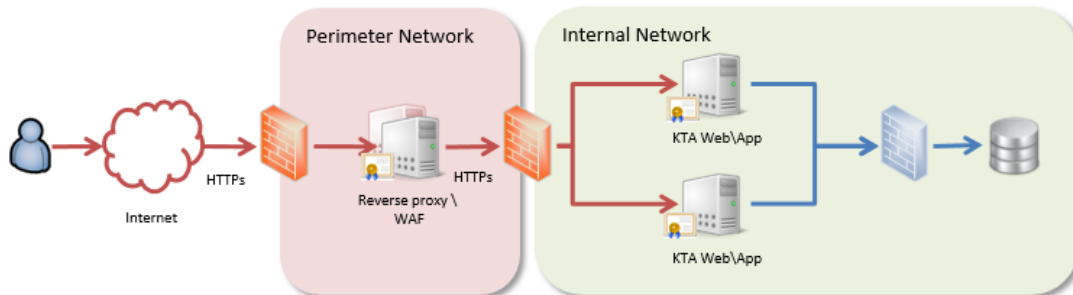
3.9.1.1. Reference Deployment 1 - Combined KTA Web and App
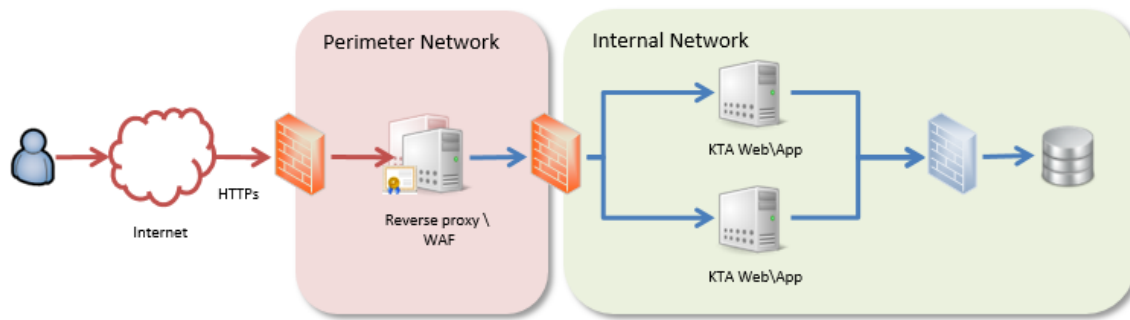
This deployment is characterised by:

- Combined KTA web\apps deployed within the Internal Network only.
- SSL terminated at the reverse proxy.
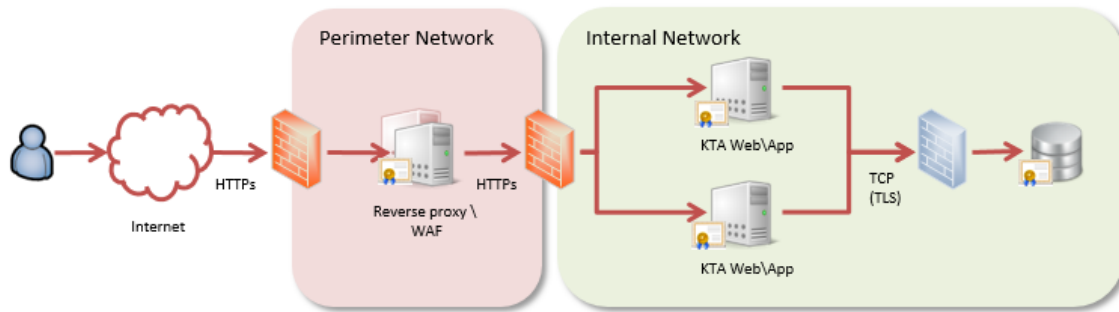- Supported deployment for an RTT server.

3.9.1.2. Reference Deployment 2 - Combined KTA Web and App

This deployment is characterised by:

- Combined KTA web\apps deployed within the Internal Network only.
- SSL decrypted at the reverse proxy\WAF and re-encrypted (i.e. bridged SSL configuration).
- Option to encrypt database traffic using TLS for database connection.
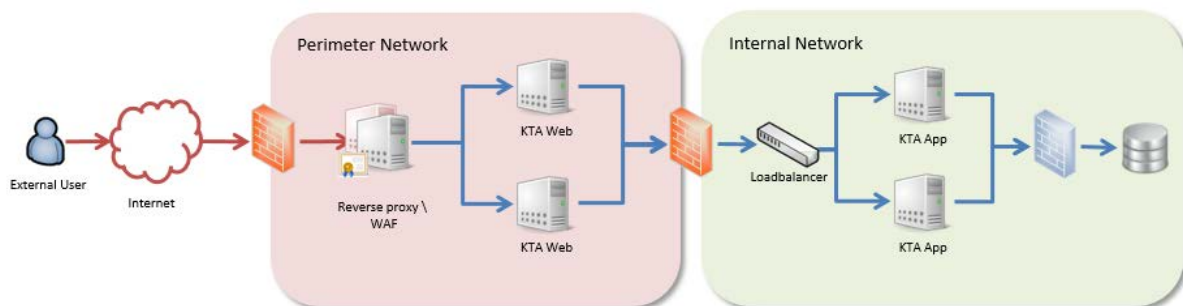- Supported deployment for an RTT server.

3.9.1.3. Reference Deployment 3 - Combined KTA Web and App

This deployment is characterised by:

- Combined KTA web\apps deployed within the Internal Network only.
- SSL decrypted at the reverse proxy\WAF and re-encrypted (i.e. bridged SSL configuration).
- TLS connection to database to encrypt database traffic
- (Option to use TDE to encrypt the data at rest).
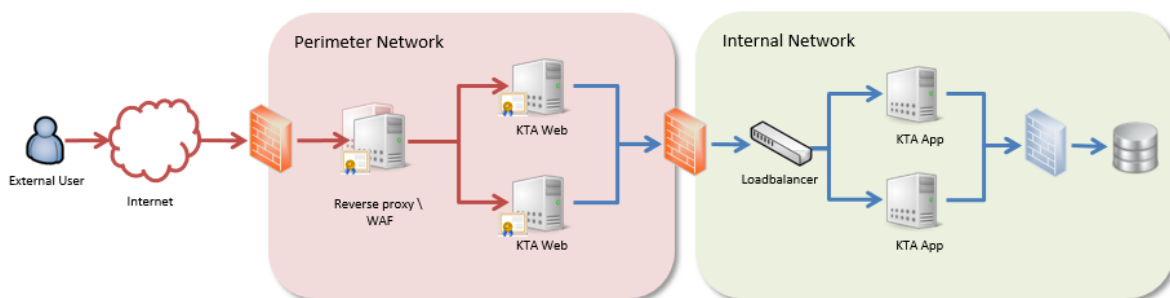- Supported deployment for an RTT server.

3.9.1.4. Reference Deployment 4 - Separate KTA Web and App

This deployment is characterised by:

- KTA web servers deployed within the perimeter network.
- KTA app servers deployed within the Internal Network only.
- SSL terminated at the reverse proxy.
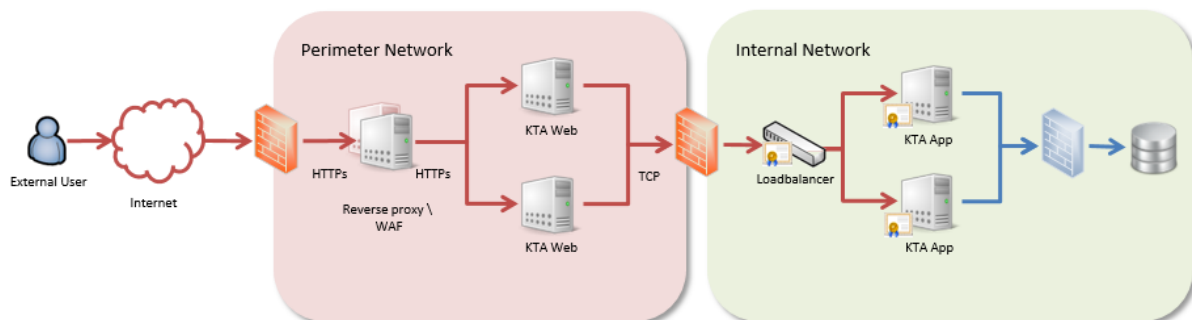- Not a supported deployment for an RTT server.

3.9.1.5. Reference Deployment 5 - Separate KTA Web and App

This deployment is characterised by:

- KTA web servers deployed within the perimeter network.
- KTA app servers deployed within the Internal Network only.
- SSL decrypted at the reverse proxy\WAF and re-encrypted (i.e. bridged SSL configuration).
- Not a supported deployment for an RTT server.

3.9.1.6. Reference Deployment 6 - Separate KTA Web and App

This deployment is characterised by:

- KTA web servers deployed within the perimeter network.
- KTA app servers deployed within the Internal Network only.
- SSL decrypted at the reverse proxy\WAF and re-encrypted (i.e. bridged SSL configuration).
- TLS connection between KTA Web and App.
- Option to use
    - TLS connection to database to encrypt database traffic
    - TDE to encrypt the data at rest).
- Not a supported deployment for an RTT server.

### 3.9.2. Reference Topologies - Insight

The following subsections depict a number of reference architectures for an internet facing deployment of Kofax Insight.

Please Note:

- The reference topologies depicted are intended to serve as a guide only; to inform design discussions surrounding the deployment options for Insight components.
- It is the responsibility of the customer to provision infrastructure to facilitate the security requirements.
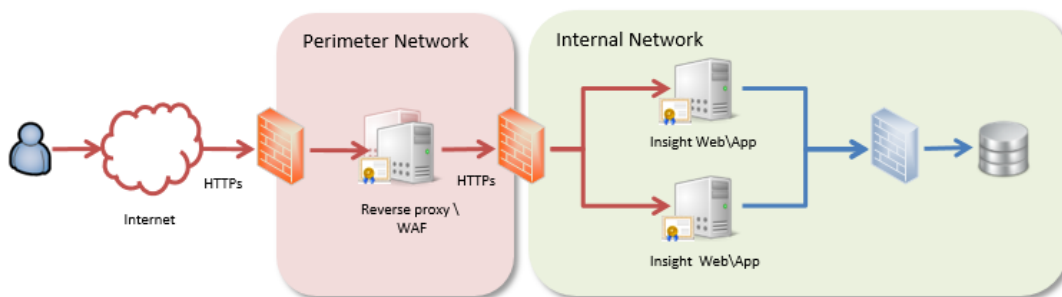
3.9.2.1. Reference Deployment 1 - Combined Insight Web and App
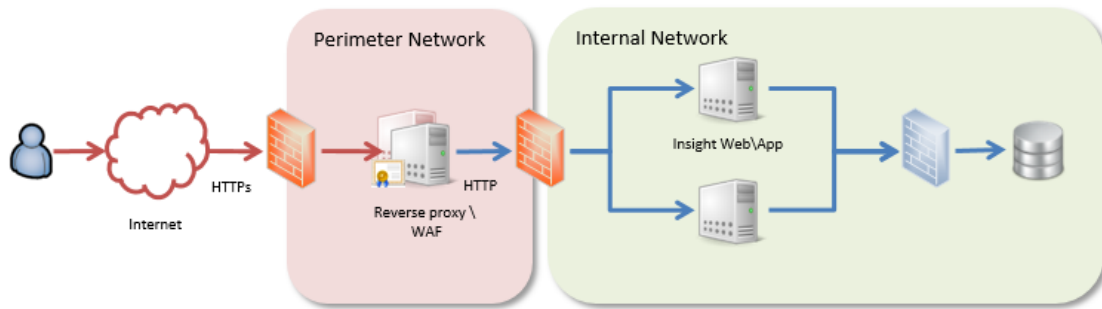
This deployment is characterised by:

- Combined Insight web\apps deployed within the Internal Network only.
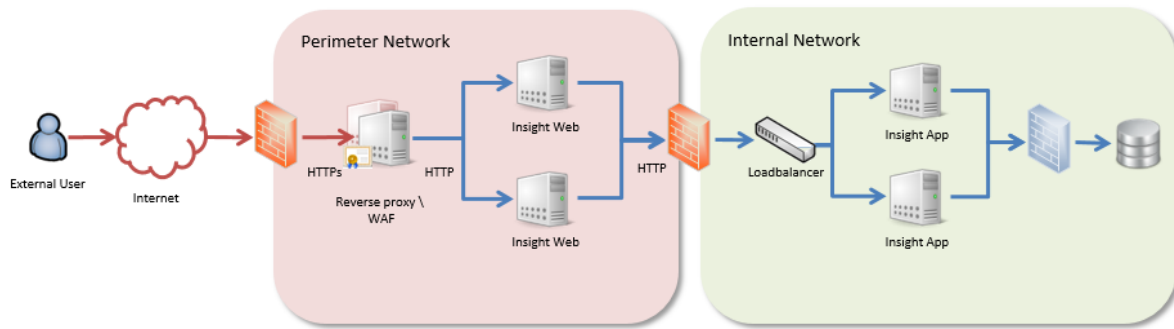- SSL terminated at the reverse proxy.

3.9.2.2. Reference Deployment 2 - Combined Insight Web and App

This deployment is characterised by:

- Combined Insight web\apps deployed within the Internal Network only.
- SSL decrypted at the reverse proxy\WAF and re-encrypted (i.e. bridged SSL configuration).
- Option to encrypt database traffic using TLS for database connection.

3.9.2.3. Reference Deployment 3 - Separate Insight Web and App

This deployment is characterised by:

-   Insight web servers deployed within the perimeter network.
-   Insight app servers deployed within the Internal Network only.
-   SSL terminated at the reverse proxy.

3.9.2.4. Reference Deployment 4 - Separate Insight Web and App

This deployment is characterised by:

-   Insight web servers deployed within the perimeter network.
-   Insight app servers deployed within the Internal Network only.
-   SSL decrypted at the reverse proxy\WAF and re-encrypted (i.e. bridged SSL configuration).
-   Option to encrypt database traffic using TLS for database connection.
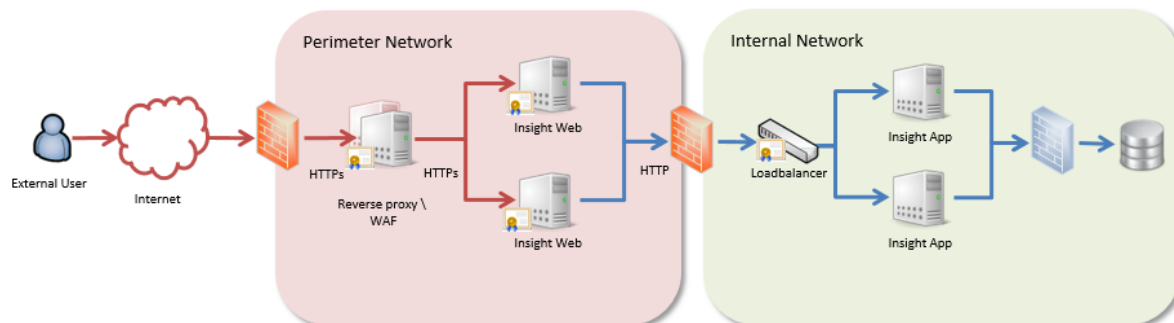
3.10. Service Accounts

Each of the services within the TotalAgility platform require a service account to run. The service account will require different privileges, depending on the role and function.

In terms of a 'standard' KTA installation, a single 'KTA' domain service account can be utilised across all of the KTA services. However, the use of discrete service accounts may be preferred for one or more of the KTA service types for security purposes (e.g. more fine grained control).

This requirement for the use of a single or multiple service accounts (as well as the provisioning of those accounts) will be the responsibility of the customer. The requirement may be based on existing generic security requirements\best practice within the customer organisation, and\or a result of project specific factors including the target deployment topology and\or individual solution requirements.

As part of a deployment planning exercise, an individual customer's security requirements regarding service identifies must be agreed.

The table below summarizes some potential configuration approaches.

Approach	Description
(1) Single KTA service account for all services (including IIS AppPools)	Simplest configuration to administer. Least secure as the single account has superset of required domain privileges.
(2) Discrete service account per service profile. E.g. - Discrete service account for coreworker and transformation (which access the KTA databases in similar manner). - Discrete service account for reporting and Insight services (access to reporting\Insight databases and read-only access to KTA core databases). - (optional) - Discrete service account for KTA web when deployed separate from KTA App (with no database access).	This represents a trade-off between administration and security: ● More administration than option 1. Less administration than option 3. ● More secure than option 1. Less secure than option 3.
(3) Discrete service account for each service type (including IIS AppPools) E.g. ● KTA_Web ● KTA_CoreWorker ● KTA_ExportWorker ● KTA_Transformation ● KTA_Reporting ● KTA_Insight ● KTA_KSMS	Most involved configuration to administer. Most secure as each account has only those individual domain privileges required to perform role.

• ...	

The table below describes the use of a single KTA service account for all applicable KTA services (i.e. option 1 above).

Server	Service(s)	Description	Service Account
TotalAgility Web Server (where deployed separately from the KTA App server)	• Internet Information Services (includes Application Pools for KTA) • Kofax TotalAgility Streaming Service	The Web Server hosts the components for the TotalAgility Website that comprises the Designer, Workspace and SDK Services. The IIS services run as the local system account. The Application Pools that are used by KTA within IIS are run as the service account.	Local System; KTA Service Account
TotalAgility Application Server	• Internet Information Services (includes Application Pools for KTA) • Kofax TotalAgility Core Worker Service • Kofax TotalAgility Export Worker Service • Kofax TotalAgility Streaming Service	The Application Server is similar to the Web Server, with services running on IIS, but it also host the Core Worker Service, which is the primary agent for business process orchestration and integration The App Server connects to SQL Server, whereas the Web Server only host the presentation layer of the application. The IIS services run as the local system account. The Application Pools that are used by KTA within IIS are run as the service account, and the Core Worker Service is run as the service account.	Local System; KTA Service Account

Kofax License Server (Primary and Backup)	Kofax License Service	The License Server is used to maintain a count of available licenses for the TotalAgility platform, for both pages and concurrent users. The scanning components of TotalAgility will communicate with the License Server to consume a page license. The web components and core worker services will communicate with the License Server to consume and release user licenses when a user connects and disconnects from a TotalAgility client, or their session times out.	Local System
Kofax Reporting Server	Kofax Reporting Service	The Kofax Reporting Server is an ETL agent used to process capture-related process information in the Reporting_Staging database and transform it into a format that can be processed by the Analytics tool. The Reporting Server Service connects to the Reporting_Staging and Reporting databases.	KTA Service Account
Transformation Server	Kofax Transformation Server	The Transformation Server is used to process capture, image transformation, data extraction and validation tasks as part of a capture process. It connects to the TotalAgility databases and the License Server. The Transformation Servers operate independently but process requests from a single pool (or specific pools if required, such as	KTA Service Account

| | | providing a separate Transformation Server to process high priority requests).

The service account on the Transformation Server requires local administrator privileges on the server. | |
|---|---|---|---|
| Kofax Search and Matching Server | • Kofax Search and Matching Server
• Kofax Search and Matching Server Browser
• Kofax Search and Matching Load Balancer | | KTA Service Account

(with logon as a service permission) |
| Kofax Analytics Server | • Internet Information Services (includes Application Pools for KTA)
• Insight Bridge Service
• Insight Scheduler Service | The Analytics Server provides both the design and runtime components for the Reports and Dashboards, hosted through IIS.

The Analytics Server connects directly to the TotalAgility databases and application databases using connection strings that are defined in the Insight Studio tool.

The service account is used to connect to the Analytics master database (to store the server settings and design settings for the various projects) and Meta and Data databases, for each solution. | KTA Service Account |
| N\A | AD Synchronisation Process (executed by KTA CoreWorker service) | The Synchronisation process used in TotalAgility that reads the Active Directory structure, targeting a specific OU to import users and runtime resource groups into the TotalAgility platform. | AD Service Account |

The information provided below is intended to supplement the information provided in the Installation Guide for the target version of TotalAgility or Insight Analytics. It provides support for situations where the server components are being scaled out into discrete server roles, for mid-large size deployments. This information is based on TotalAgility 7.1/7.2 and verification should be done against your target version before installing the software.

Local Administrator privilege is required on each of the servers to install the software, typically done using a named user account within the target domain, and differing permissions for each service account to operate the system.

The permissions required by each service account are defined as per the server role as follows. Note that the service accounts on the Application Server (TotalAgility and Analytics), Reporting Server and Database Server also require privileges in the product and application databases as shown.

The table below outlines the required Windows and database permissions per KTA service

Server	Windows Permissions	Database Permissions
TotalAgility Web Server	The right to enumerate against the domain to validate network user accounts.	None
TotalAgility Application Server	"Log on as Service" rights; The right to enumerate against the domain to validate network user accounts.	db_datareader, db_datawriter and execute in the TotalAgility product databases
Kofax License Server	N/A - services run as the "Local System" account.	None
Kofax Reporting Server	"Log on as Service" rights	db_datareader, db_datawriter and execute In Reporting_Staging and Reporting databases.
Transformation Server	Local Administrator privilege	db_datareader, db_datawriter and execute in the TotalAgility product databases
Kofax Search and Matching Server	Must be a member of the "KSMSAdministrators" group (for operation) - either local or domain	read permission in any databases used as a

	group	datasource
Kofax Analytics Server	"Log on as Service" rights	Create, drop and modify index, tables and stored
		Procedures for Insight databases (MetaDB, DataDB and AdministrationDB).
		db_datawriter and execute in the TotalAgility product and custom application databases

Note: If the installer user or account does not have permission to create the Product databases for either TotalAgility or Analytics, these will need to be pre-created prior to installation. Follow the steps in the "Installing the Databases" section of the TotalAgility Installation Guide and the "Insight Installation Manager" section of the Insight Installation Guide for details on pre-creating the required databases.

4. Installation Best Practice

This section provides guidance and best practice for installing and configuring the TotalAgility platform, including preparing the servers for installation, options for deploying the software components, and post-install configuration and tuning of the software.

4.1. Installation Prerequisites

The prerequisites required on each server for TotalAgility and Analytics differ based on the server role and the version of Windows Server being used. The prerequisites required for each version are provided in the Installation Guide for each product. This document illustrates installation requirements through use of KTA 7.2 and Insight 5.2. The prerequisites for each server role are shown in the sections below, split into Windows Roles and Features and Third-Party software requirements. These settings are based on an installation on Windows Server 2012 (R2).

It is important to check these prerequisites against the Installation Guide for the target version of the software, and confirm these prior to performing the product installation. It is preferable for a customer to take ownership and responsibility for configuring the server and installing any additional third-party software prior to performing the TotalAgility and Insight product installations.

The Appendix titled "TotalAgility Installation Prerequisites" contains generic steps to configure the Windows Roles and Features in Windows Server 2012 (R2) and install the third-party software components.

Note: When applying the Windows Roles and Features to the target server, it is possible to export a template of these settings as an XML file for scripted installation on other servers. It is recommended to perform a one-time manual configuration of these settings for the first server installation, e.g. in the Development environment, and export the settings so that they can be applied automatically to subsequent server installs, for the current environment and higher environments.

4.1.1. Windows Roles and Features

Server	Windows Roles and Features
License Server	N/A
TotalAgility Web Server	- Server Roles: - Application Server - Web Server - Features: .NET Framework 4.5: - ASP .NET 4.5 - WCF Services: - HTTP Activation

	- Application Server Role Services: .NET Framework 4.5 - Web Server (IIS Support) - Windows Process Activation Support: - HTTP Activation - Web Server Role Services: - Security: - Basic Authentication - Windows Authentication **Note:** If you install IIS before you install the .NET Framework, register the .NET Framework with IIS. Run "aspnet_regiis -i" from the command prompt.
TotalAgility Application Server	- Server Roles: - Application Server - Web Server - Features: .NET Framework 3.5 .NET Framework 4.5: - ASP .NET 4.5 - WCF Services: - HTTP Activation - Application Server Role Services: .NET Framework 4.5 - Web Server (IIS Support) - Windows Process Activation Support: - HTTP Activation - Web Server Role Services: - Security: - Basic Authentication - Windows Authentication **Note:** If you install IIS before you install the .NET Framework, register the .NET Framework with IIS. Run "aspnet_regiis -i" from the command prompt.
Reporting Server	N/A
Transformation Server	N/A
Kofax Search and Matching Server	N/A
Analytics Server	IIS - The following IIS settings must be turned ON: - Web Management Tools - IIS Management Console - Application Development .NET extensibility - ASP.NET - ISAPI extensions

	○ ISAPI filters
	● Common Http Features
	   ○ Default Document
	   ○ HTTP Errors
	   ○ Http Redirection
	   ○ Static Content
	● Health and Diagnostic
	   ○ HTTP Logging
	   ○ Request Monitor
	● Performance Features
	   ○ Dynamic Content Compression
	   ○ Static Content Compression
	● Security
	   ○ Basic Authentication
	   ○ Client Cert mapping Authentication
	   ○ IIS Client Cert mapping Authentication
	   ○ Request Filtering
	   ○ URL authorization
	   ○ Windows Authentication
	
	For Windows 8 and Windows 2012+ Server HTTP Activation needs to be turned ON under the WCF Services features for the 4.5 .NET Framework.

4.1.2. Third-Party Software

Server	Third-Party Software
License Server	● .NET Framework 3.5
TotalAgility Web Server	● .NET Framework 4.5 and later
TotalAgility Application Server	● .NET Framework 3.5 SP1 (used by Import Connector) ● .NET Framework 4.5 and later ● Microsoft ODBC Driver 11 for SQL Server ● Microsoft Command Line Utilities 11 for SQL Server ● Microsoft SQL Server 2012 Native Client ● Microsoft Distributed Transaction Coordinator (MSDTC) is required when the databases are split
Reporting Server	● .NET Framework 4.5 and later
Transformation Server	● Python 2.5.1 is required to use the A2iA Zone Locator, and must be installed manually. **Note:** Other prerequisites are installed automatically by the Transformation Server installer if they are not present on the machine.

Kofax Search and Matching Server	Prerequisites for KSMS are included with the product installer.
Analytics Server	.NET Framework 4.5.2 - Microsoft ODBC Driver 11 for SQL Server - Microsoft Command Line Utilities 11 for SQL Server - Microsoft SQL Server 2012 Native Client

Note: Windows Server 2012 comes pre-installed with .NET Framework 4.5.

4.2. **Service Accounts**

Service accounts required for installation (referenced above) should normally be decided and documented as part of the initial architecture of the platform for the customer. Installation generally cannot proceed without these, and there is typically a lead-time to have these type of accounts provisioned, especially within a customer's Production system, so it is advised to establish agreement on these early.

4.3. **Installation Order**

In a complex environment, where multiple servers are being deployed for the TotalAgility platform, the servers should be installed in the following order, though some servers can be installed in parallel, subject to the dependencies noted below:

Step	Component	Dependency	Comments
1.	SQL Server		Database server needs to be provisioned with the requested disk space, and access granted to the relevant service accounts, administrator and installer.
2.	License Server		Install primary and backup License Servers. License is activated by the Application Server installation.
3.	TotalAgility Application Server	2, 3	Install the primary Application Server and create the required databases and schemas. Install additional Application Servers and connect to the existing databases. A load balancer is required to

			support multiple Application Servers.
4.	TotalAgility Web Server	4	Install each of the Web Servers and connect to the Application Server(s) directly or using a load balancer. A load balancer is required to support multiple Web Servers.
5.	Reporting Server	4	Primary and secondary Reporting Server, configured as active/passive. Can be co-located with the License Server or Application Server.
6.	Transformation Server	4	Each Transformation Server can be installed independently.
7.	Kofax Search and Matching Server	4	Install each Search and Matching Server and configure clustering.
8.	Apply applicable Service Packs or Fix Packs to the TotalAgility servers	1-7	Apply a service pack and/or fix pack for the installed version of TotalAgility, to each server component.
9.	Analytics Application Server	3	Install the primary Insight Application Server, create the required databases and schemas. Install additional Application Servers and connect to the existing databases. A load balancer is required to support multiple Insight Application Servers.
10.	Analytics Web Server	9	Install each of the Insight Web Servers and connect to the Insight Application Server(s) directly or using a load balancer.

			A load balancer is required to support multiple Insight Web Servers.
11.	Apply applicable Service Packs or Fix Packs to the Analytics Servers	10	Apply a service pack and/or fix pack for the installed version of Insight Analytics, to each server component.
12.	Perform post-installation configuration of the Analytics server	11	Perform post-installation configuration of Analytics using the Administrator client, including license activation, configuration of the authentication and users and roles.
13.	Deploy KAFTA/KIAFTA solution.	12	Deploy the KAFTA/KIAFTA solution to the Analytics Server and configure the solution for the target environment.

The detailed steps for installation can be found in the Kofax TotalAgility Installation Guide, the Kofax Search and Matching Server Installation Guide, the Kofax Insight Installation Guide and the Kofax Analytics for TotalAgility Administrator's Guide.

4.4. **Scripted vs Wizard Installation**

Each component of the TotalAgility platform can be installed using a silent installer, in some cases coupled with a configuration file to define the installation options. For both the TotalAgility platform and Insight Analytics, some level of post install configuration is required. This post-install configuration includes actions such as:

- Activating the license;
- Configuring the system settings for email, session timeouts etc.;
- Configuring resources, roles and access restrictions;
- Integrating with third-party systems, such as Active Directory, SharePoint etc.

These topics are discussed in more detail below in the section titled "Post-Install Configuration".

Generally speaking, for a low volume or low complexity solution, where the TotalAgility platform will only be installed on a small number of servers or environments, it is easier to perform a manual install using the installer wizard.

For more complex deployments, with multiple environments and options to scale out the individual servers based on server role, it is recommended to script the installation.

4.5. **Post-Install Configuration**

This section provides a list of items that should be considered for configuration after installing the base software components. This includes:

- Validation of the installation to ensure all components are working as expected;
- Configuration of the TotalAgility System Settings:
    - Session Timeouts
    - Email Settings
    - Access Restrictions
    - Archival of completed jobs
    - Retention policies for design artifacts, users, audit log and documents
    - User login and password policy settings
    - Regeneration of the System Session ID
- Creation of Users and Groups in TotalAgility;
- Configuration of Business Calendars;
- Applying default settings for exception processes;
- Synchronisation with Active Directory using the System Job (if required);
- Integration with third-party systems;
- Post-installation configuration of Insight Analytics:
    - Authentication Settings
    - User and Role Mapping
    - Integration with TotalAgility
- Encryption of the TotalAgility configuration files;
- Tuning of dependent software components, such as IIS;
- Disabling KTA Services that are not required e.g. Export Service, Reporting Service etc.

These topics are developed further with some specific points discussed below.

4.6. **Installation Validation**

Having performed an installation of KTA components, the installer should seek to verify the installation. This verification activity (and any necessary troubleshooting) should be an integral part of any installation project.

These post installation checks should include confirmation that:

- The Installation program did not log any errors
- The Designer can be launched and that authentication occurs as per expected configuration (e.g. Windows Auth, manual login etc.)
    - And a sample of the designer screens can be accessed without error.
- The Workspace can be launched and that authentication occurs as per expected configuration (e.g. Windows Auth, manual login etc.)
    - And a sample of the workspace screens can be accessed without error.
- A Job can be started via the workspace and listed in Job view and workqueue (where applicable)
- Each service can start
- Each service can start without logging any eventlog errors\warnings.
- The relevant service is able to process work without error i.e.
    - Core Worker can process automatic activities
    - Transformation Server can process capture activities
    - Reporting Service is performing ETL on reporting events
    - Etc.

- Generated files assemblies can be created on the file system (business rules, sync processes, document generation

In a scaled out deployment,

- The initial installation of each server type should be verified before proceeding to install additional instances.
  - This enables issues to be identified and the installation\configuration procedure to be adjusted prior to attempting additional set-up.
- Where required, services can be disabled on 'all other servers' to validate the behaviour of a service(s) on a specific machine.
- Where a load balancer is in use, load balancing behaviour should be configured (having verified individual nodes as outlined above).

Where an installation/upgrade involves re-pointing services from one KTA database to another, then the originating database should be taken offline to ensure that no services are still pointing at the original location.

4.7. IIS Configuration

Internet Information Services (IIS) on Windows Server 2012 provides the administrator with the option of timing out a worker process that is idle for a specified period of time. This is a good option for sites that are not accessed very often because it frees up system resources when the site is idle. The disadvantage is that the next time the site is accessed, the user will need to wait for the worker process to start again. The default value for idle time-out is 20 minutes.

Idle time-out can be helpful in the following situations:

- The server has a heavy processing load.
- Specific worker processes are consistently idle.
- No new processing space is available.

For systems where the processing is on-demand and SLAs are critical, it may be appropriate to remove or increase the idle time-out to keep the application active and in-memory.
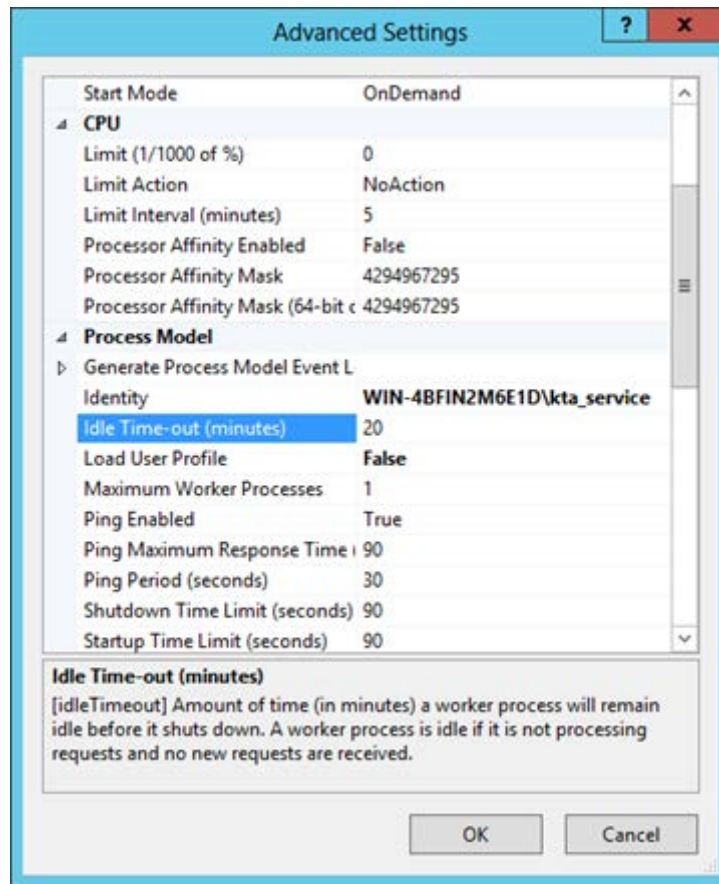
You can change the Idle Time-out to 0 to disable, or increase the time-out value.
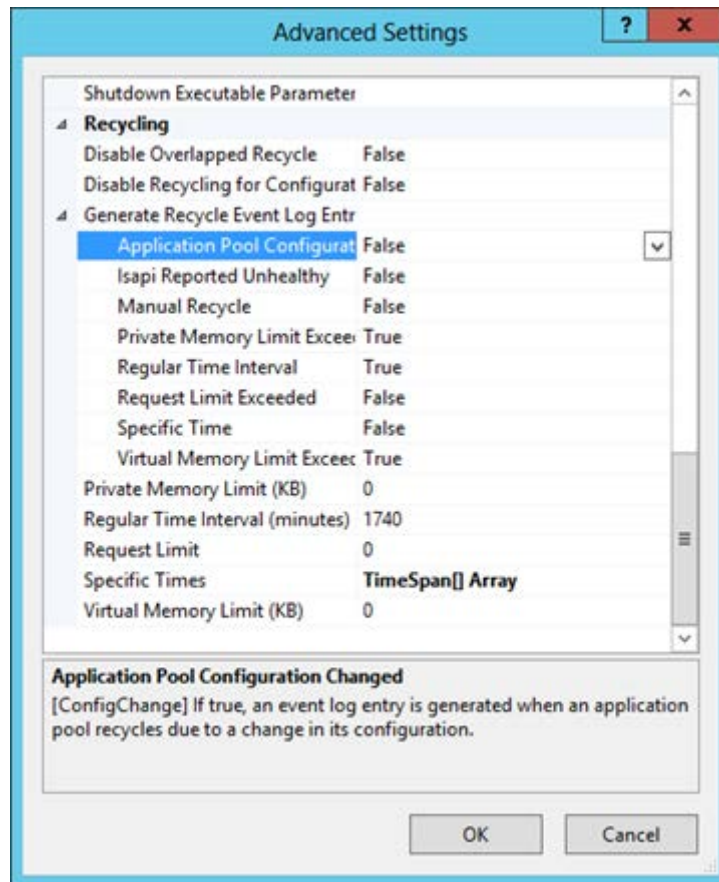
You can change the Recycling Settings, for example changing the time to execute to be "out of hours" so as not to impact users that are accessing the system during the day.

The default settings are to recycle every 29 hours. You can set it to recycle each day at a specific time, e.g. 23:30.

Note: This guidance assumes that IIS is being dedicated for TotalAgility, and any changes to the Application Pool recycling will not affect other sites or applications within IIS.

4.8. Transformation Server Caching

Transformation Server downloads the classification and extraction projects in memory the first time that they are used on a server. This can cause a small delay the first time that the projects are used after restarting the service/server, due to time required to download the project.

Under normal circumstances this is not an issue, as the image processing activities are performed asynchronously as part of a business process. It may become an issue where synchronous processing of document classification or extraction is required, such real-time processing from Mobile Capture, where the user is waiting for a response back from the application.

In KTA7.3 classification and extraction project can be pre-loaded. In other versions there is a workaround to pre-cache the classification and extraction projects, by creating a Windows executable, written in .NET, to call the Business Process web services and trigger a business process to perform classification and extraction for the target document class(s) using sample

documents. This executable can be scheduled to run as a Windows Task, triggered when the Transformation Server service is restarted.

The process for configuring this trigger in Windows is described in the Appendix titled "[Pre-caching Classification and Extraction projects on Transformation Server](#)".

5. Application Security

It is important to understand the options and steps necessary to secure access to design-time tools and components being developed, e.g. for situations where multiple streams of developers are working within the same instance of TotalAgility and security separation is required to keep the streams of work separate. Similarly, it is important to understand how to provide secure access to run-time tools and components, including access to create/modify users, roles and access, and access to forms and processes within a TotalAgility application.

5.1. Identity Management/Access Management

Identity Management/Access Management is a broad topic that defines how users are identified within an organisation, and how access to resources and applications within an organisation are granted and controlled. The level of management required for TotalAgility depends greatly on the size and scale of the customer, and the level of maturity of their IT processes.

There are some standard ways for provisioning users within the TotalAgility application, and granting access, including:

- Manually provisioning users and groups using the TotalAgility Designer or Workspace in the target environment.
- Synchronising users and groups from Active Directory or a native LDAP server using the Active Directory Synchronisation system process that is provided out-of-the-box with TotalAgility.
- Implementing Federated Security to integrate with an authentication provider to provision users on-demand using a security token from the authentication provider.

Each of these configurations are covered in the TotalAgility Help, available from within the TotalAgility Designer or the Help within the TotalAgility Workspace (for manually creating and managing users and groups).

There are other ways to manage users and groups in TotalAgility, depending on the customers IM/IAM system and the level of integration required. This includes scripted creation of resource users and groups by parsing an export from an external system, or allowing an Access Management system to connect to TotalAgility via the SDK web services to automatically create resource users and groups.

5.2. Design-time Groups vs Run-time Groups

When using Active Directory to synchronise users and groups in each TotalAgility environment, it is recommended to separate the groups used to assign access to the TotalAgility components at design-time from the groups used to assign user access at run-

time, for the purpose of deploying the design-time groups between environments, and allowing the run-time groups to be created separately in each environment.

This is particularly important when different Active Directory domains are used to synchronise users and groups for each environment, as the groups synchronised for each environment will be uniquely identified by their GUID within AD. There is a requirement for the design-time groups to retain the same GUID in each environment.

The design-time and run-time groups need to be mapped one-to-one, with the design-time groups created in the Development environment and assigned access to each of the components of the solution, e.g. Category, Forms and Processes. They are then deployed to each subsequent environment as part of a TotalAgility package. The run-time groups are created for each environment by the Active Directory synchronisation process.

5.3. **User Authentication**

TotalAgility provides the following user authentication mechanisms out-of-the-box:

- **Basic or Application Authentication** - this requires the usernames and passwords to be defined for each user and stored within TotalAgility, with the password encrypted. A user is authenticated using an application login form. The credentials from the login form are authenticated against the stored username and password;
- **Windows Authentication** - the Windows user credentials are sent to the Web server from the client browser, based on the logged in user. IIS validates the credentials against the network domain, and passes through to the application. TotalAgility uses the username of the logged in user to create a session for the resources with a matching username. If the user does not exist within TotalAgility an error message is displayed.
- **Federated Security (KTA 7.2 onwards)** - Federated Security uses a trusted third-party authentication provider to authenticate a user and provide a claims token to identify the user and provide some context of their role or persona. TotalAgility acts as a claims-aware application that can integrate with the authentication provider and interpret the claims token for a user. The first time a user logs into the application using federated security a user profile is created for them in TotalAgility based on the context provided in the token, e.g. group membership, email address etc.

The out-of-the-box authentication mechanism can also be customised, for example to integrate with a SSO provider and use the TotalAgility SDK to create a session for the user once they have been authenticated. Refer to the UserService in the TotalAgility API documentation for more information.

5.4. **Single Sign-On (SSO)**

The diagram below outlines a common mechanism for integrating KTA Forms into a SSO environment (where Federated Security is not being employed).
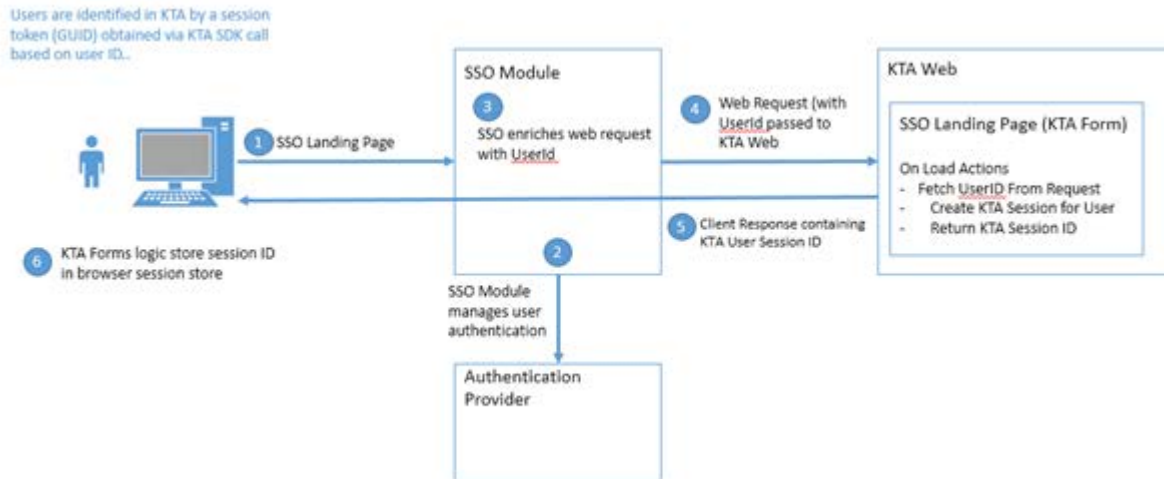
This mechanism allocates responsibilities as follows:

- The SSO Component is responsible for
  - User authentication.
  - Providing an authenticated user's identity to KTA (i.e. username string)
- KTA is
  - Agnostic to the authentication mechanism(s) being employed.
  - Responsible for commencing a KTA user session.

Note: Regardless of the authentication mechanism employed (with or without SSO), KTA requires that users are defined locally within the KTA user repository to facilitate authorisation and activity assignment (See the KTA User Repository section later in this document).

5.4.1. SSO Module

The SSO Module (e.g. SiteMinder, WebSEAL etc.), is responsible for performing the user authentication, and presenting the userid of the authenticated user to KTA to enable KTA to commence a user session for the identified user.

The user id may be passed to KTA in a number of different ways. The mechanism chosen will determine the implementation approach adopted within KTA.

5.4.1.1. QueryString Parameter

The SSO module may enrich the request URL to include an additional QueryString parameter designating the requesting userid.

The target SSO Landing Page (KTA Form) can be configured to expect a UserID initialisation variable

5.4.1.2. HTTP Header Token

The SSO module may add a 'userid' token to the inbound request header.

The SSO Landing Page (KTA Form), can utilise a simple .NET method to extract the UserID.

5.4.1.3. KTA Service Endpoints

To facilitate communication with the KTA webservice endpoints, both IIS and the individual endpoint(s) must be configured appropriately.

- IIS must be configured to allow anonymous access.
- The HTTP service endpoint (used by KTA Forms incl. Workspace) must be configured to allow anonymous requests
  - The configuration snippet below presents the configuration required on the webHttpBinding.

```xml
<webHttpBinding>
    <binding name="WebHttpBinding_Service"
        <security mode="Transport">
            <transport clientCredentialType="None"/>
        </security>
```

6. Auditing and Logging

This section provides an overview of the auditing and logging that is performed by each of the components within the TotalAgility platform. At a high-level, application logging is written to the Audit database for TotalAgility, which is enabled by default. System-level logging is written to the Windows Event Logs on each of the servers.

6.1. Application Audit Log

The audit trail is created by default in the Main TotalAgility database. It is recommended to separate the Audit database into its own database within a Production environment.

By default, audit trail entries are kept indefinitely. It is good practice to configure a retention policy for the audit trail, to remove the audit events after a certain period of time. The retention period is dependent on the customer's requirements, and will be impacted by their SIEM (Security Information and Event Management) policies. The retention settings can be configured in the System Settings section in the TotalAgility Designer, see below.
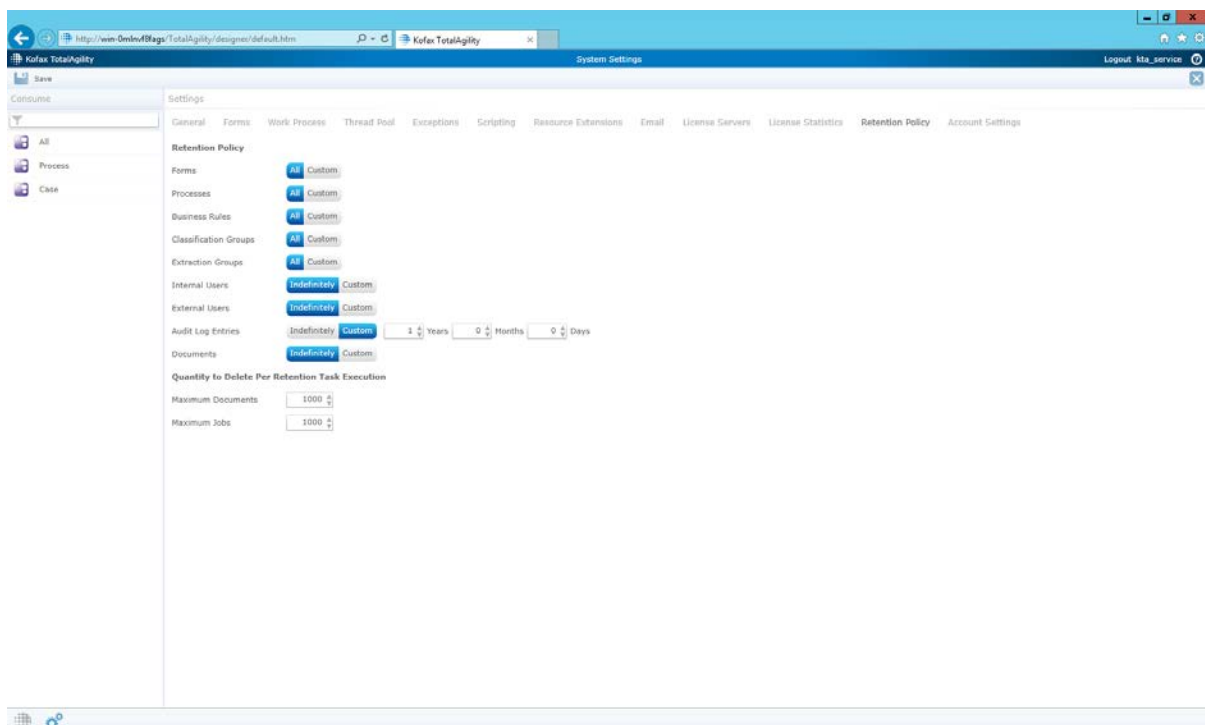
Large customers that operate within a strict regulatory environment, such as government, banking and finance and healthcare, typically have strict SIEM policies and will have tools for centrally managing application auditing and logging. For these customers, they can integrate their SIEM systems directly with the TotalAgility Audit database, to export the audit trail to a centrally managed system, where it can be monitored and analysed to detect security or policy breaches. Once this information has been exported the audit trail entries can be removed from the TotalAgility database.

SIEM policies typically require system events to be logged and monitored. The events created by the components of the TotalAgility platform are described in more detail in the section "Windows Event Logs".

6.2. IIS Logging

The table below outlines the components deployed into IIS and the associated log paths.

Server	IIS (default) website name requiring IIS logs to be monitored	IIS Website log paths for SIEM team to configure monitoring
Web Server	TotalAgility	Default, e.g.: %SystemDrive%\inetpub\logs\LogFiles
Application Server	TotalAgility	Default: %SystemDrive%\inetpub\logs\LogFiles
Analytics Server	Analytics	Default: %SystemDrive%\inetpub\logs\LogFiles

6.3. Windows Event Logs & Text Logs

Refer to the section on Monitoring and Resilience below for details on Windows Event Logs and Text Logs.

7. On-Premise Multi-Tenancy

On-Premise Multi-Tenancy was released with TotalAgility 7.1. It provides a way of provisioning a single set of servers to support multiple, discrete instances of TotalAgility. It uses a shared license but allows allocation of user, image capture and transformation rights individually to each Tenant.

It is important to understand if OPMT is the right choice for your solution, as it adds complexity to the architecture, design, development and support of the platform. OPMT should be considered for cases where the customer requires separation of the application among multiple tenants while still using shared infrastructure. It is possible to share a single License Server for multiple instances of the TotalAgility platform, so multi-tenancy is not required for this use-case, although it would be required if you want to control or limit the allocation of licenses for particular instances.

Key Differences between On-Premise and On-Premise Multi-Tenancy

The key differences between On-Premise and On-Premise Multi-Tenancy deployment models include the following:

- OPMT requires an additional Tenant Administration component, and associated databases, to store the Tenant configuration, platform and Licensing configuration;
- Each Tenant requires an Integration Server to be deployed within their data center and connected to the Application Server(s) in the hosted OPMT platform. This is used to execute activities within the business process that integrate with their back-end systems, such as Document Generation, Export and Email or File Import using the Import Message Connector;
- Insight Analytics and the KAFTA solution is not yet multi-tenant aware, requiring a separate Analytics Server to be deployed for each Tenant;
- The SDK API cannot be integrated into the Business Process or Forms using the application DLLs, it requires integration by consuming the associated web service endpoints.

Please refer to Product documentation on OPMT or consult with Professional Services consultants if you require additional information or assistance.

8. Development Approach

When developing a solution using KTA, the development team must decide between the use of:

- a single shared KTA development environment; or
- multiple shared KTA development environments for distinct project capabilities
- e.g.
    - Team A working on Classification and Extraction
    - Team B working on Process Implementation
    - Team C working on Integration of CCM, Kapow (Robotic Process Automation), etc.

- individual developer environments
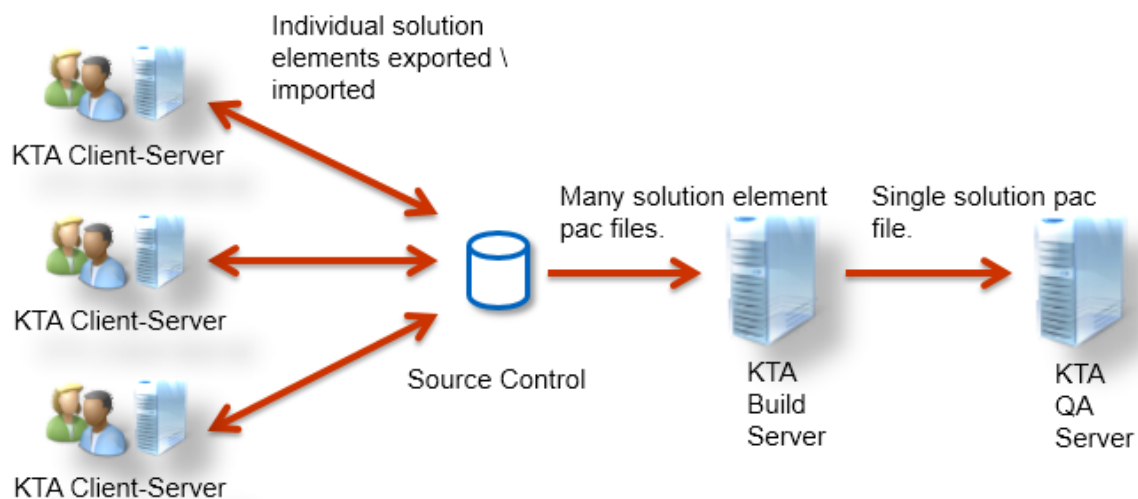
The decision to adopt one approach versus the other is likely to be driven by:

- Connectivity
    - Offline working will require individual developer environments, in which case SDLC (software development lifecycle) procedure must be used to properly govern the synchronization of solution assets.
- Customer compliance \ Internal procedures
- Team Size

In any case, the developers should place all work under source control, and follow standard practices accordingly.

9. TotalAgility Forms vs Custom Forms

The TotalAgility Forms Designer and Forms Engine provide an easy way to build forms using a visual design tool with a drag-and-drop style configuration, coupled with auto-generation of forms based on the target business process and activity. This includes capture-enabled forms, for web capture, document review and validation activities.

As a forms builder technology, the visual drag-and-drop capability reduces the complexity of building user interfaces. The trade-off in utilising such an environment, however, is that the capabilities of the tool are, by definition, a cut down version of a full scale development environment. In other words, TotalAgility Forms is not a replacement for a fully featured development language or associated development environment such as ASP.NET MVC and Visual Studio.

Depending on the solution requirements, there may be a need to develop some or all of the components using a fully featured development language, integrated with TotalAgility using the SDK API.

10. Managing or Exchanging Data

General rules

The only data maintained within the process should comprise:

- Data required to drive the process e.g. utilised in a decision node, or passed to a third party API call.
- Extended Work Queue Definition data

All application data should be maintained in an application (i.e. custom) database.

Work Queue Definition Data

Work Queue definition data is likely to be duplicated between the process and the application database, so either

- the data will be read-only when the process commences (preferred if possible), or
- a synchronisation step must occur when the shared data is changed either within the process or from within the application.

Benefits of Externalising Data

Keeping data external has significant benefits:

- Maps are easier to create\maintain

- The less process data required, the less variable configuration required when developing maps.

- The resultant process is more resilient to data change, as the map is likely to only track key index data e.g. Customer ID. Any changes to the associated data will therefore have no impact on the process.

- Data is easier to access, for reporting or other purposes
  - Variable data held within the process can only be accessed via the API on a job by job basis.
  - Data held in the database can be accessed by any SQL tool

System of Record (SOR)

KTA is not intended to be a primary system of record. It orchestrates the execution of business processes and for that purpose stores process information while a process instance (e.g. a borrower's loan application process) is "in process". Important information that should be persisted when the process is complete should be held in an appropriate system of record (e.g. customer database or ECM/records management system).

- The data being processed by KTA may already reside in a suitable data repository which can be accessed directly by KTA.
- The data may be retained long after the business processes have completed, and it is therefore best that this data lives outside of the orchestration product.

TotalAgility does store process design information and can act as the system of record for these designs. KTA's architecture is such that these process designs are held in a database (a MSSQL db).

TotalAgility generates audit logs, recording who did what when etc. These also are stored in a database and these "records" can be persisted by appropriate use of audit log and archive settings.

11. Capture/Extraction Best Practice

This section will explain general considerations and best practices on implementing a Capture Project in KTA.

It will not go into low level details of how to extract fields or Transformation Designer technical locator implementation details. For more information on Developing with Transformation Designer, please refer to available materials including product documentation and Education class materials.

11.1. Customer Engagement Considerations

There are several important considerations when engaging in a project that includes Classification and Extraction.

11.1.1. Expectations Management

It is important from the very start of the project, to establish with the customer a set of realistic expectations of what can and will be achieved through Classification and/or Extraction.

Specific areas in which clear expectation setting helps preempt later issues include the following:

Document Review

Customers should establish an early understanding of how much document review is required for Classification and Separation Projects. Some customers require advice in respect of the role and purpose of document review in order to plan for this task in their operation.

Validation

Similarly, the extent of Validation required, or how Validation works for Extraction fields is sometimes not well understood and should be clarified.

It is sometimes useful at an early stage to conduct a demonstration of this Validation process with some relevant customer documents.

Straight Through Processing

Documents can only be enabled to go "straight through" if the customer provides supportive business rules that allow the majority of fields to pass validation rules.

It is useful, for the purpose of managing expectations, to demonstrate that default behaviour is that some Validation will be required. A customer may believe that Validation will be a rare event and that everything else will go "straight through". The demonstration will help the customer understand that, as a general rule, the more fields available, the more likely a document needs to be validated.

Classification or Extraction % Success Rates

There are three major expectations that should be addressed early in the implementation:

1. The measurement metric. Many metrics are available but a customer must choose the measures most appropriate in their business. For example, extraction rates are available at a document field level, yet while useful this may not be the best metric for the business. Focusing on productivity improvements such as documents per operator or field throughput per operator is often a better and more relevant metric. Customers sometimes want to measure on a batch level or a whole of document level, and they may have heard percentage rates from other teams or other customers (that may have been measured differently). This sometimes means expectations are unachievable. Ensure that the customer and all team members are educated that extraction rates are on a field level, and in addition that this is often not the best measure to use.
2. Labor savings: This factor is often a significant aspect of the business justification for a capture implementation and significant benefits are available. However, expectations should be clearly set and it is therefore important to ensure that reasonable assumptions are made based on savings available through efficiency in document review, validation automation, etc.
3. False positives. While Review and Validation try to detect low confidence to flag these to the operator, there will always be some fields on some samples that fall into the category of "False Positives" where the system is confident, yet the OCR, Extraction or Classification is incorrect. The customer must allow adjustment of confidence levels for these, and for "critical" fields it is sometimes worthwhile having that one particular field always stop to be checked, or place strong validation rules on it to automate such checking.

11.1.2. Document/Forms Design

Early consideration of the following factors can improve efficiency and effectiveness of document-based business process design.

Barcodes

Three main recommendations should be taken into account in respect of using barcodes productively:

1. For purpose of document separation, if barcodes are on the first page of a document and have a known content. This is even more useful if the documents have a fixed page length.
2. For Classification, the classification can be embedded in a barcode.
3. Extraction assistance. If a dynamic barcode is used, you may include some details on the extraction content or something that will allow you to perform a database lookup. This may help you with Validation and/or the Extraction as well.

Form Design Guidelines

Kofax has published some form design guidelines, for more details refer to the document "How to design optimal forms for Kofax Transformation".

11.1.3. Sample Collection

Optimized capture performance requires good use of sample documents. Plan for this early in the implementation project plan. Inform the customer project team, as close to the start of the project as possible, how many documents must be collected.

Classification Projects

200 Samples per Document Type is a good number to start with, where the documents are variable. Fewer may suffice if barcodes are successful, or if the documents are always printed and always very much the same.

A separate set of samples per input channel is required as well, including:

- Scan / Hard Copy
- Fax
- Email
- Photograph / Mobile
- Original electronic PDF
- Office (Word/Excel)

The larger the number of input channels, the more sample documents are needed.

Extraction Projects

The number of samples that are required for extraction projects varies with the type of data that will be extracted.

As a general rule:

- Structured Documents (forms) - 10 fully filled samples for development per form, and 200 forms for benchmarking. The 200 benchmarking forms should have realistic data, including fields that may not be filled.
- Unstructured Documents (e.g. Invoices) - 500 samples per document type for development purposes to train the system with an initial knowledgebase, plus 200 samples for benchmarking.

Where there are multiple channels, additional samples may be needed per channel including:

- Scan / Hard Copy
- Fax
- Email
- Photograph / Mobile
- Original electronic PDF
- Office (Word/Excel)

11.1.4. Customer Assistance in Golden Set and Benchmark Preparation

The customer must be informed early on, to make staff available for purposes of data entry (and Lasso) of documents for the Benchmarking Process. It is important for the customer to be involved in this process as the benchmark may not be correct if the data is validated by a consultant that does not have business knowledge.

11.1.5. Statement of Work Considerations

There are a number of factors to consider for creating an effective Statement of Work (SoW) that results in both a happy customer and a successful project.

Whether you are a project manager planning the implementation of TotalAgility to meet requirements of your "customer" in the business, or you are a third party working to implement a solution on behalf of a customer, you should create a Statement of Work or similar document that outlines clearly what work you agree to do and what you agree to deliver for your customer.

Require Signoff of Golden Set Before Development

The Golden Set is critical during development and testing of the Transformation Project. If the Golden Set is a moving target and keeps changing during development, this will result in delays, growth in development effort, and ultimately a less happy customer.

Getting the signoff ensures that the customer has considered and confirms that these are the correct documents and correct formats against which the solution should be developed and measured.

Require Signoff of Benchmark Before UAT

The benchmark shows the level of success per field that is achieved on untrained documents, based upon the golden set provided.

The consultant may tune the system up to a cost effective point during development, but there are potentially infinite number of tuning/benchmark cycles that can be done throughout a project - the only limit being the budget and timeframe.

Therefore it is very important that a point of compromise and acceptance is reached on the extraction rates at the end of development before user acceptance testing is entered into. If this is not done, UAT can cause re-entry into the development phase for additional tuning cycles and wasted development effort for minimal payoff. This should be done before handing the project to the customer, not after.

Require Signoff of System Detailed Design (SDD) Before Development

As above, SDD should not be constantly changing during development, and an agreed scope in detail upfront ensures what is developed is correct.

Do not Pre-Define Extraction or Classification Percentages

Doing the development and running a benchmark is the only way to know what success you can achieve.

Customers require the BEST result that a tuning and benchmarking process creates, not a Pre-Defined result that is made up before conducting this process.

Therefore it is in the interest of all parties to not pre-set any percentage levels, and instead define the process that will be followed to achieve best results for minimum costs.

11.2. Capture Workshops

When preparing for implementation of a capture solution, involve customer staff in Capture Workshops so that requirements are well understood at the start of the implementation project. This section recommends the structure of these workshops and the required attendees and also touches on the role of document analysis, system detailed design and "golden sets" in the workshop context.

11.2.1. Structure

Workshops should aim to capture information on all of the following topics:

- All required KTA topics such as infrastructure, environments, process flow, security, operators, exceptions etc
- Golden Set and Benchmark Set
- Image Cleanups and Conversion (file formats)
- Input Requirements (KIC, Scan, Email, SDK etc)
- Output Requirements (KTA output activities OR Export Connector)
- Document Classification Strategy, and type breakdown
- Document Separation Strategy
- Forms Design and Barcodes
- Extraction Documents
- Fields for All Documents
- Formatting rules for each field
- Validation (business) rules for each set of fields
- Validation Screen layouts
- Integrations Required

11.2.2. Attendees

- Customer Business Analyst or Subject matter expert
- Operator/Validation team leader
- Customer Project Manager (Business)
- Customer IT Project Manager
- IT personnel required for integrations

11.2.3. Document Analysis

A document analysis at the workshop can help in setting expectations and getting a general idea of how the data will be extracted. However it is best to avoid locking down the exact methods of data extraction at this stage, as multiple methods may be used during development. The important thing at this stage is to capture exactly what needs to be done for each field, not how it's going to be done.

11.2.4. System Detailed Design

A system detailed design should be produced based upon all the topics covered in the workshops.

The customer should signoff this is correct before commencing any development. This is very important to prevent excessive iterations of development, and delays to the project timeline.

11.2.5. Golden Sets

The Golden set is detailed in a separate section, but it is important to discuss it during the workshops and get agreement on what documents are going to be provided, and a deadline.

11.3. Input Considerations

11.3.1. Scan

In KTA there are alternative methods to scan physical documents that need to be considered on their merits for the customer. The methods that can be considered are:

- Direct scanning using the KTA Scan Form with Plugin
- MFD Scanning to Folder (then pickup via KIC or other method)
- MFD Scanning directly to KTA (for supported MFDs only) - see specific MFD documentation that comes with KTA documentation pack.
- Kofax Express that exports to KTA. Benefit here is provision of a thick client.
- Separate Kofax Capture system that exports to KTA. Benefit here is provision of a thick client and the full features of both KC/KT and KTA in one multi-step process.

11.3.2. Import and KIC

KTA provides a flexible means of importing images and documents. VRS is linked to the Image Processing Activity in KTA. This activity takes place after process initiation, and you can therefore use any method to get the documents into the process and still have the benefits of VRS. For example, you can use SDK Job Service to create the job, .NET activities to create Capture Documents or Folders, Script Activities, Initialise from Scan, RTT, Mobile Capture, etc.

11.4. Using synchronous activities in the SDK

If synchronous calls will be used SDK, for example to exploit real time transformation (RTT) to return extract information quickly to a scanning device, the implementation team needs to plan appropriately by answering key project design and planning questions:

- Will synchronous calls be used or asynchronous
- If a synchronous call is to be used, how will the unattended activity of document deletion be handled, without deleting the document that the calling program requires. If Validation is required, this would not be a problem as the call will return at Validation. However if Validation can be skipped for a straight through process, the Delete Activity may need to be moved to a separate job and have a precondition event added to prevent it occurring too early.
- If using synchronous calls, is RTT required for the use case. Note that for Synchronous calls, the Transformation service will be notified to process a job immediately instead of waiting to Poll. It also load balances these notifications automatically using the TRANS_SERVER_INSTANCE table. This table is updated and maintained automatically when services are started or stopped.

11.5. Output Considerations

KC/KT implementations typically use an "Export Connector" or "Release Script" via the Export Module to output documents and information to archives, repositories or other systems. Sometimes this export function would also be achieved using a customer module or scripting.

KTA on the other hand is a lot more flexible. While KTA has similar "Export Connector" capabilities (please refer to the Compatibility Matrix on the internet for an up-to-date list of

which connectors are compatible with KTA), KTA also has a range of other methods to provide output to other systems.

Export or KTA Native

KTA includes export connector capabilities and these can be used where the project's needs are limited to the connector's straightforward functionality. These capabilities are legacy capabilities originally made available for KC/KT implementations. Where their out of the box capabilities are adequate for the projects purpose, they can be used and will minimise cost and development time since they are "ready to use".

However, using a KTA native method is usually preferred and has the advantage of being customisable, and allows business processing / workflow operations to occur as well.

Possible activities to consider for outputting data from KTA are:

- Kofax Export Connector
- .NET activity
- Web Service
- Data Access
- Email
- SharePoint
- CMIS
- TRIM
- Script Activity

11.6.   Image Processing and VRS

11.6.1. VRS Profile

- Always create a named VRS profile, instead of using the default. This allows the same Platform to be used for other projects, each project then having its own VRS profiles.
- Prefix the VRS profile name with the category name (since VRS profiles don't currently support categories)
- For mobile capture or photographed documents, a fixed paper size such as A4 or Letter can help standardise input. This is prudent because every camera has a different resolution and aspect ratio.
- 300dpi is the accepted standard and is widely acknowledged within Kofax to provide the best OCR and Extraction results for Transformation Server
- Black and White (enhanced) is the recommended color setting, for reasons of file size and quality OCR results. Only choose color if it is a specific customer requirement to retain the color.
- 'Delete blank page' capability can be very useful especially in cases of duplex scans

11.6.2. Image Processing

The main purpose of image processing is to send the image through the VRS profile and run any cleanup and conversion.

It is recommended to convert all images to TIFF, if they need to be used in Validation directly. For example for Online Learning, you cannot Lasso on PDF documents.

It is also recommended to use renditions in order to retain the original document when converting to TIFF. However this must be discussed during design phase, as sometimes it is detrimental to retain the original (for example when redaction is required, original may not be allowed to be stored).

11.7. Use of Folders for captured documents

As a general rule, a Folder should be used for the Capture documents to allow flexibility to collect multiple documents. Where document separation is required, a Folder must to be used.

In some use cases single document processing may be the focus, however even in these cases a Folder can still be used.

11.8. Composite Activity or Individual

A Composite Activity can combine all of Image Processing, Classification, and Extraction into a single activity.

This can increase the performance significantly in the Transformation Server, rather than launching 3 separate activities that have to retrieve the documents each time.

Generally a Composite Activity can be used to improve performance except where:

a) Document review is needed between classification and extraction. In this case you may still get partial use out of a composite activity by bundling Image Processing and Classification into a Composite Activity.
b) Other operations need to occur between processing, or logic operations need to occur. For example checking results of each step with decision nodes before proceeding to the next.

An alternative to using a Composite Activity, is to spawn separate processes to continue processing without waiting for all constituent activities of a Composite Activity to complete.

11.9. Shared or Non-Shared Groups

Shared extraction groups have the following advantages:

● Can import from a previous KC/KT project
● Can export the project to a backup or to another system
● Can share scripting and code between the classification and extraction
● Can test Classification and Extraction in Transformation Designer with one click instead of working in 2 separate projects.

Note that although you can import/export shared groups, you cannot import over the top of an existing group that is tied to a Process. For this reason, exporting an extraction group is not considered sufficient backup because there is no easy way to restore it. Instead, it is better to backup the associated Process itself. To restore an extraction group would require you to delete the original extraction group, import it again, and re-map everything including formatting and validation rules and mappings within the process.

Separating Classification and Extraction Groups has the benefit that multiple developers can work at the same time. For example, you may assign a classification group to 1 developer, and then multiple extraction groups (say 1 per document) to other developers.

11.10. Formatting Rules

Formatting rules that are setup on the extraction group in KTA will be executed automatically in both Transformation Server, and in Validation.

It is not necessary to add extra formatters into the Transformation Project itself, unless they are needed for helping the extraction (e.g. extracting dates with month names that format the month to a digit, or formatters that feed into the Invoice locators).

11.11.   Validation Rules

Validation rules should be developed as reusable rules, in a generic manner, such that a single rule could be used on multiple fields.

11.12.   Skipping Document Review

The diagram below provides an object model that may be used in determining whether or not to skip document review or Validation.
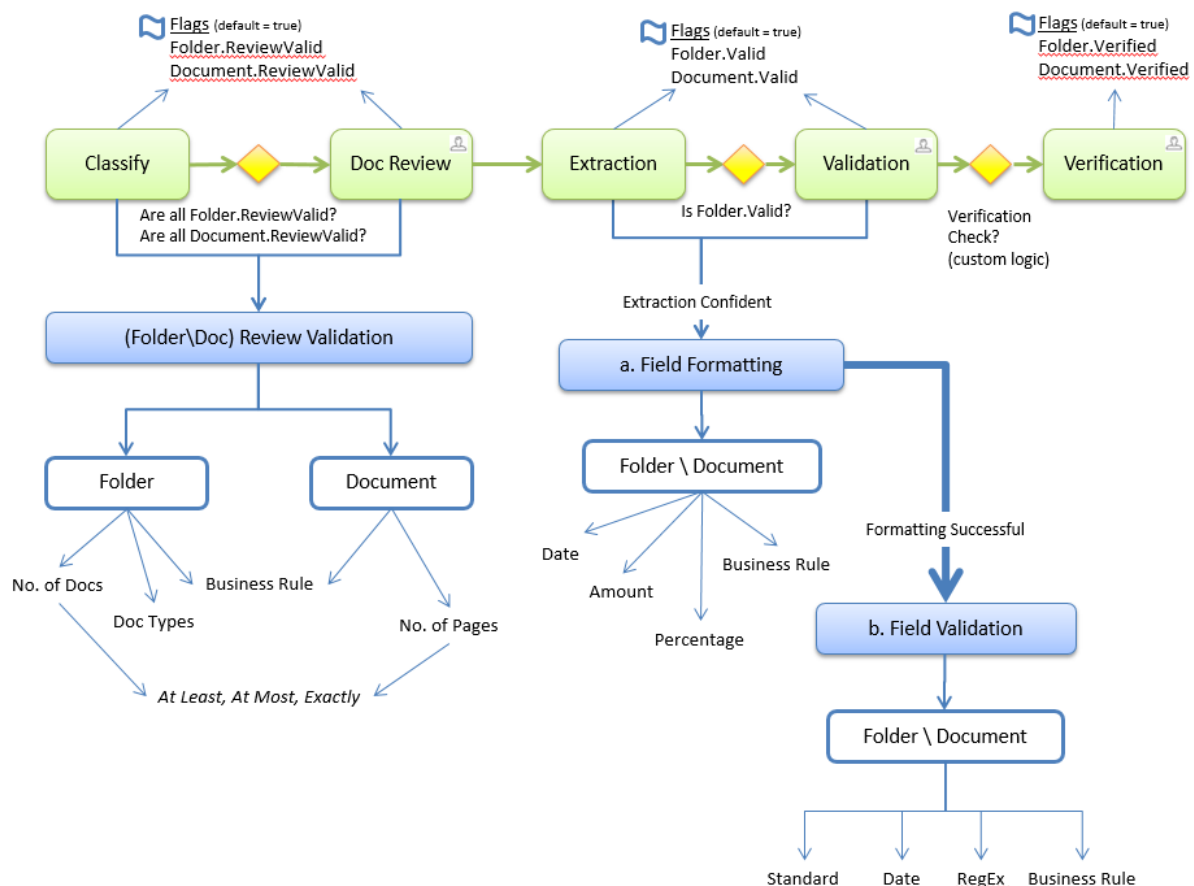
11.13.   Skipping Validation

Skipping Validation can be done in a similar method to skipping document review, however for documents that have critical fields or a large number of fields it is not recommended to skip Validation.

When setting up a process to skip Validation, it is worthwhile trialing the system initially with Validation 'on' to gain some confidence about the level of false positives that may occur. It is

good practice to implement any skipping of Validation as a Boolean switch in a variable so that it can be turned on and off easily by the customer.

11.14. PDF Conversion

PDF Conversion is optional and is dependent on customer requirements for their output system.

The advantages of PDF over TIFF however are:

- More widely accepted standard of file format in viewers and document management systems
- The ability to define metadata on the PDF document and populate it from Variables:
  - Title
  - Author
  - Subject
  - Keywords
  - Application
  - Producer
- The ability to embed OCR Text inside the PDF
- PDF/A compliance (archiving standard)
- With the availability now of PDF Compression, PDF size difference is no longer a disadvantage compared to TIFF.

The licensing for PDF Text and PDF Compression are included as part of the standard KTA license (whereas these are separate licenses in KC/KT).

11.15. Deletion

The Capture delete activity should always be added to any normal Capture Process to ensure the database does not grow excessively. If this is not done old images will be left in the TotalAgility database and will cause the database size to be larger than it need be.

Normal procedure is to delete when the business process is completely finished with the image, i.e. after Validation, Export etc have all completed.

11.16. Backing up your Transformation Project

The best way to backup your transformation project, with all mappings, validation rule assignments and formatting assignments, is to Export the whole Process.

Although extraction groups and transformation projects keep a version history, this history is not usable. You cannot revert to a previous version, nor can you delete the current version to revert. Therefore backup the whole process to backup your extraction group and transformation project.

Exporting a File Based Project does not allow re-import into the same extraction group. If you delete the extraction group, all mappings and references will be lost.

Exporting to a file based project does, however, have value is some uses - you can go to the filesystem folder and copy/backup all your knowledge base *.kb* files, and backup your training set documents. These can be imported into other projects or other environments.

11.17. Deployment Considerations

11.17.1. Dictionaries

Transformation keeps an imported copy of the dictionaries, and therefore will still operate after a deployment even if the original/higher environment does not have these dictionaries.

However it is a better practice to have a location with all dictionaries on all environments so that they can stand independently if "hotfixes" to the code are needed. This is also important if dictionary contents are environment dependent. For example dictionaries on production may contain production data that is not available on lower environments.

11.17.2. Retaining Knowledge Bases & Online Learning

When deploying to production it is advised to backup the production knowledge bases and online learning. The following process is suggested to allow this:

a) Open Transformation Designer on the Production system
b) From the documents toolbar, Download new samples
c) If multiple downloads are necessary, keep downloading until all samples are successfully downloaded
d) You should now be on the view for "New Samples"
e) Select the import button to import the new samples into a Training Set
f) Click on "Extraction Set" and you will see a sub-set. Note that if other subsets have not yet been compiled into *.kb* knowledgebases these may also need to be done (i.e. if someone has imported new samples on production before)
g) Expand the document class on the left, and on all trainable locators (TGL, IGL, OGL, AGL) double click on them and perform the below process on all of them
h) Select knowledge base tab
i) Select Create to create a knowledgebase
j) Use a naming convention to include the document type, kb type and YYYMMMDD. For example Payslips_Generic_20160101
k) After all knowledgebases are created, you may disable the new extraction sets or delete them
l) From the main menu, select "Export to File-based Project"
m) Browse to the exported folder, and take a backup copy of all the files in the Knowledgebase Folder.
n) Deploy your new version
o) Open the new version after deployment, and import/assign the kb files. Now your new version will have all the online learning that was collected in production.

12. Process Modelling Best Practice

12.1. Business versus System Exceptions

Exception handling can be divided into two main categories:

- Business Exceptions
- System Exceptions

It is important when designing exception processes to consider the type of exception and thus the required behaviour of the exception process.

12.1.1. Business Exceptions

This grouping includes exceptions relating to cost or timing overruns. The exception handling process is therefore a business response to a business issue. The participants (if any) in such a process are therefore business users.

In general, the response to these exceptions usually includes one or more of:

- Update a Job State
- Send Email Reminder to resource(s)
- Send Email Reminder to Group\Resource Supervisor
- Assign an activity to the Supervisor (prompting them to take action)
- Re-assign an overdue activity to another resource (e.g. Supervisor)

12.1.2. Resource Handling

Where the response to a business exception involves logic associated with activity assignment (e.g. reassign activity to supervisor), it should be noted that the target activity may have multiple resources assigned (including a combination of individual resources and groups). Any processing of the resource information may therefore have to handle single and multiple resource assignments (where applicable).

12.1.3. System Exceptions

This grouping includes exceptions thrown by the system e.g. automatic activity failures. The exception handling process is therefore a technical support\operations response.

These processes should therefore be in line with company policy regarding system support. Participants in this process are likely to be technical support personnel (or systems). Relevant business users may also be alerted to the issue.

12.1.3.1. Exception Handling

Implement a strategy for handling job suspensions based on system exceptions

- Implement a **server level** exception handling process.
  - Approach may include one or more of:
    - x automatic retries (potentially with sliding retry interval)
    - notify (e.g. e-mail, SMS) on x failures
    - re-notify on
      - one hour since last notification
      - x failures since last notification
    - manual activity assigned to administration
    - escalation notification \ activity assignment
  - Consider volume exception scenarios
    - e.g. webservice endpoint unreachable suspending 10K jobs

■ i.e. don't send individual e-mails on each suspension!
- Process Level (only where required)
  - ○ Override the generic handling approach where specific handling required.

12.2. Exceptions and Triggers

Triggers and Exceptions represent two mechanisms for handling activity due date and job duration overruns. It is important to be aware that the two approaches exist and to use the approach most appropriate for a given scenario.

This section details the two approaches and provides guidelines on deciding the appropriate approach to use.

12.2.1. Activity Due Date Exceeded

There are two mechanisms for raising Activity Due Date Exceptions

- Exception Evt10013
- Activity Due Date Trigger

Activity Due Date Triggers pass the same information (to a configured process) as Evt10013. As a general rule, use the Exception Evt10013 unless you identify a specific need to use the Activity Due Date Trigger.

The Exception Evt10013 provides a mechanism for defining a default (server level) exception and overriding this where required at an individual process level. This approach is the simplest to set-up and maintain, where activity due date exceptions are handled commonly at the server and\or process level.

As a guide, use Activity Due Date Trigger when

- You want to catch due date exception before they happen (pre-exception). Exception Evt10013 only fires when the due date has been exceeded. An Activity Due Date Trigger can be configured to fire prior to the due date. Zero or more triggers can be defined for a single activity. This provides a mechanism for pro-active management of activities which look like they might miss their target due date.
- You want to handle post-exception due date triggers. Once an Activity Due Date has been exceeded and the Exception Evt10013 has fired, the only way to trigger further exceptions for the activity is to change the due date (and reset the exception). With an Activity Due Date Trigger, the due date can remain the same (as originally set), and one or more Activity Due Date Triggers defined to fire after the Activity Due Date has been exceeded (by setting a trigger time relative to the due date).
- You want to handle due date triggers differently at the activity level. With Exception Evt10013, details of the activity that caused the exception are passed to the exception map. The map can be designed to handle the exception differently dependent upon the source activity. With Activity Due Date Triggers, a different exception process can be executed based on the source activity.
- You want to handle due date triggers differently based on the triggers relationship to the due date i.e. pre-exceeded, exceeded or post-exceeded. Different processes can be executed based on which of the triggers has fired.

Note: It should be noted that unlike process level exceptions which override server level exceptions (where specified and applicable), triggers will not override the associated server level (or process level) exceptions. Care must therefore be taken if a combination of the two approaches is utilised.

12.2.2. Job Duration Overruns

There are two mechanisms for raising Job Duration overruns

- Exception EVT0005
- Job Duration Trigger

EVT0005 passes the JobID to the escalation process. In addition to JobID, Job Duration Triggers pass the Estimated Finish Time. As a general rule, use the Exception EVT0005 unless you identify a specific need to use the Job Duration Trigger.

The Exception EVT0005 provides a mechanism for defining a default (server level) exception and overriding this where required at an individual process level. This approach is the simplest to set-up and maintain, where job duration overrun exceptions are handled commonly at the server and\or process level.

As a guide, use Job Duration Overrun Trigger when

- You want to catch job duration exceptions before they happen (pre-exception). Exception Evt10005 only fires when the duration has overrun. A Job Duration Overrun Trigger can be configured to fire prior to the overrun datetime. Zero or more triggers can be defined for a single process. This provides a mechanism for pro-active management of jobs which look like they might miss their target completion date.
- You want to handle post-exception duration triggers. Once a Job Duration Overrun occurs and the Exception Evt10005 has fired, the only way to trigger further exceptions for the job is to change the target duration (and reset the exception). With a Job Duration Overrun Trigger, the target duration can remain the same (as originally set), and one or more Job Duration Overrun Triggers defined to fire after the Job Duration Overrun has occurred (by setting a trigger time relative to the expected duration).
- You want to handle target duration overrun triggers differently based on the triggers relationship to the target completion date i.e. pre-overrun, overrun or post-overrun. Different processes can be executed based on which of the triggers has fired.

Note:

It should be noted that unlike process level exceptions which override server level exceptions (where specified and applicable), triggers will not override the associated server level (or process level) exceptions. Care must therefore be taken if a combination of the two approaches is utilised.

Milestone triggers offer an additional SLA based event mechanism at a process level.

There is no equivalent server level exception associated with a milestone overrun.

12.3. Activity Notification

The Activity Notification facility within KTA provides a mechanism for sending a notification e-mail to the resource(s) associated with an activity which becomes 'pending' (i.e. available to be taken and completed).

When considering the use of this facility, it's important to consider

- The nature of the process

- The relationship of the resource associated with the activity to the process.

General rules

Participants who are actively engaged with a process (or set of processes), are likely to regularly interact with a Work Queue, listing activities assigned to them (or related groups) in priority and due date order. In this case, the work queue should be a sufficient tool for viewing and managing the progression of work items. In this instance, an e-mail notification adds no value. In fact, being bombarded with emails is likely to annoy the user.

Infrequent participants who do not regularly participate in a given process are less likely to monitor a work queue regularly. In this instance, there is significant value in notifying the user that they have a work assignment.

In some instances, high priority activities benefit from the notification mechanism (even when assigned to frequent users). The user is notified immediately of the pending activity, allowing an organisation to reduce\eliminate the potential time lag between the activity becoming pending and the user becoming aware of it.

Note: The Email Notification feature is implemented as a System Process Map (i.e. SYSTEM Process Email Activity). It is therefore possible to modify this map to achieve the desired notification behaviour e.g. send an SMS message as well as\instead of e-mail.

It should be noted however that:

- The initialisation parameters cannot be modified (as the notification feature has been built to pass a specific set of parameters)
- This process will be utilised by activity notification feature system wide, i.e. any changes will impact all activity notifications.

12.4. Use of (.NET) Script Nodes

Script Nodes provide an additional option for implementing custom logic (in addition to the creation of a .NET component(s)). They offer a convenient mechanism for executing custom code without the need for a separate development environment (e.g. Visual Studio) in which to implement the logic.

They are useful when e.g.:

- Calling a method which returns a complex object (i.e. class instance). The script can be used to process the return object and populate the necessary process variables as appropriate.
- Wrapping a number of calls to expose a single 'service' call e.g. logon, do something and then log-off
- Stubbing a method call for demonstration or testing purposes

When developing custom code, the developer needs to make a conscious decision to use a script node or a custom component. If a pre-existing custom component(s) have already been developed and used, then adding functionality to the component(s) is likely to be a better choice that creating a new component or using a script node.

Scope

If the functionality is required outside of the process(s), then the functionality should be placed in a component to make it easily accessible from other components\systems.

Ease of development

Developing .NET code in a fully featured development environment e.g. Microsoft Visual Studio, is easier.

From within KTA, it's easier to map parameters to method calls on a custom component (drag and drop) than it is to get and set parameter values within a script.

Tip: Even when using a script node, it is often better to write the code using a fully featured development environment e.g. Microsoft Visual Studio, where the developer can avail from the rich toolset provided. Once developed and tested, the code can then be pasted into the script node.

Ease of testing

A script node needs to be tested as part of a process test. A component can be tested independently of the process platform.

Ease of deployment

If the new functionality results in the creation of a new component(s), then deployment procedures must cater for the deployment of the component(s).

In addition, source control and configuration management processes must manage the new component(s).

Script Failures

Failures in a script node will result in suspended jobs. Bugs in script nodes would require correction and the re-deployment of the process map (new version). Bugs in a custom component however, can be corrected and a new version of the component released. The suspended processes could be re-activated to utilise the corrected component.

As a general rule, it is best to avoid the use of script nodes in preference to the development of a .net component, except for the implementation of a small amount of simple logic.

Where a .Net component is already in use, then all logic should be implemented within the component for simplicity and consistency.

12.5. Use of Activity Types and System Variables

When developing process maps, it's important to be aware of the system features provided to make interaction with KTA easier.

KTA provides a number of mechanisms to simplify process development where the process needs to interact with KTA. These include

- System Variables

12.5.1. System Variables

Care should be taken to ensure that, if a system variable is available that it is used in preference to an API call to retrieve the same information. The system variables provide a convenient (and reliable) mechanism for retrieving key system\process data easily.

The use of these system variables results in a process map which is easier to understand and maintain.

E.g. where a process name is required within a process,

- define a process variable of type system variable SPP_PROCESS_NAME, and map accordingly within the process, as opposed to calling e.g. JobService::GetMinJobDetails via a .NET node to retrieve the same information.

12.5.2. Activity Types

KTA Provides a number of activity types which provide a convenient mechanism for performing common process tasks. These include:

- Job Owner Activity – For setting the Job Owner
- Resource Activity – For retrieving information relating to a specified resource
- Job Variable Activity – For retrieving process data from a specified job.

These activities provide a quick, convenient and tested mechanism for fetching\setting common data as described above. These activities should therefore be used in preference to making direct API calls on the KTA Platform.

The consistent use of these activities (where applicable) results in processes which are easier to understand and maintain.

12.6. Variable Naming

It is important that variables are named correctly when initially created.

- Within the KTA Designer, the variable display name is shown when mapping variables (and the display name is always editable).
- When invoking the KTA API and passing variables, the VariableID is used.

Variable names cannot be changed once the variable has been created. To change a variable name, the existing variable must be deleted and a new variable added. In addition, if the variable is already associated with one or more activities, it must be disassociated from all the activities before it can be deleted.

Common Usage

Any given process map is likely to include common variables, i.e. variables that are used for the same purpose in many maps. For example, variables used for dynamic allocation are

utilised in many processes. It is worth defining an agreed common format for these variable names to achieve consistency and aid maintainability.

For example, variables holding dynamic resource XML string could be called:

- RESOURCE_ID_XML
- RESOURCE_ID_DYN

Specific instances could be:

- MANAGER_ID_XML
- SUPERVISOR_ID_DYN

12.7. **Process Commenting**

Use process descriptions and annotations to improve readability and maintainability of processes.

13. Work Prioritisation

13.1. **Workqueue Interaction**

Workqueue presentation and interaction is an area where solutions can run into difficulties by implementing approaches that are inefficient (for the end users), introduce unnecessary system load and\or do not scale.

When implementing a workqueue approach consider user interaction carefully by asking questions that uncover user requirements. For example:

Q. Does the user need to see a workqueue and have the ability to select from a no. of activities, or do they need simply to be given the next activity (the latter requirement would indicate a need for a 'Get Next' implementation).

Q. Does the user (or a subset of users) need to have the flexibility to use a GetNext mechanism and <u>also</u> have the flexibility to access a workqueue view?

<u>If presenting a workqueue, select an appropriate no. of activities to present to the end user.</u>

- Return the minimum number of activities necessary to enable a user to select an activity to progress (e.g. 10-50).
- Avoid returning large no's of activities as this:
  - puts unnecessary load on the system & consequently impacts system performance negatively
  - reduces the UI responsiveness for the user
  - is not a practical way for a user to select work
    - work will be returned in priority order, so a user should normally select the first activity on the workqueue list (in which case there is no need to present 10's or 100's of work items on a workqueue list).

<u>Avoid custom sorting</u>

The workqueue is designed to return work in 'priority order' based on either job level or activity level due dates and assigned priority values (1-100). Users are usually expected to work on items in the configured priority order.

Attempting to apply a custom sort to a workqueue requires all of the work items to be returned so that the custom sort can be applied to the entire set. This is not a scalable approach, so consider carefully your requirements and try to avoid this approach.

If sorting is requested:

- explore the underlying business requirement (i.e. why is a user looking to sort the work in an unprioritized order?)
- Determine the best approach to meet the underlying requirement:
  - Consider use of filters (on same view or separate view)
  - Consider how work is prioritized and determine if adjusting the job\activity priorities and\or due dates to reflect the underlying need will address the requirement.

13.2. Work Assignment

Avoid users having to return work to the workqueue as they are unable to progress a particular activity due to either:

- Inappropriate skills
- Dependency on another event(s)

Inappropriate Skills

Create an appropriate resource assignment model and assignment approach such that activities are assigned only to people with the appropriate skills and knowledge to complete those activities.

- Create separate groups signifying different skills and assign activities accordingly.
- Use skill levels (where appropriate) to differentiate resources in the same group in conjunction with process skill settings).

This avoids the inefficient behaviour of a user taking an activity, assessing the activity to discover they are not capable of progressing the activity and then returning the activity to the workqueue for a more appropriate resource to progress.

Dependency on other event(s)

Workqueues should only contain activities that can actually be progressed. Where an activity is dependent on another event(s) having occurred, the process map should be designed such that that activity is not made pending until those dependencies have been satisfied.

This avoids the inefficient behaviour of a user taking an activity, assessing the activity to discover that there is an outstanding dependency and then therefore returning the activity to the workqueue.

For example, if an activity requires that a 3rd party is contacted, yet that 3rd party is known to be unavailable for 2 weeks, then the associated job should be placed on hold (with appropriate mechanisms to automatically reactivate), rather than the user cancelling the activity and then seeking to avoid the activity which remains on the workqueue.

13.3. Job/Activity Due Date and Priority

In addition to the prioritisation of manual activities, be mindful that automatic activities are also subject to prioritisation.

Where job level due date and\or priority value are being used for prioritisation, ensure that ALL (asynchronous) process definitions are considered and the job due date and\or priority are set appropriately.

Where activity level due date and\or priority value are being used for prioritisation, ensure that ALL automatic activities (in asynchronous processes) are considered, and the activity due date and\or priority are set appropriately.

In addition, consider threadpool configuration (threadpools, threads and prioritisation) where there may be a requirement to throttle the processing of certain processes or activities to prioritise work under heavier loads.

Ensure the job/activity, threadpool and engine settings match the required prioritisation as this may significantly impact system performance, particularly under heavy load conditions.

![Lexmark Enterprise Software logo]

14. Configuration Management

All solution artefacts should be managed in an appropriate configuration management tool (e.g. TFS).

The project structure adopted within the configuration management tool should be consistent across all projects.

14.1.   Project Structure

A suggested project structure is as follows:

- \<Customer>
    - \<Project>
        - Database
            - \<database>
                - Tables
                - Stored Procedures
                - Views
                - ...
        - KTA
            - Forms
            - Processes
            - Business Rules
            - .NET
            - Data
                - Classification groups
                - Extraction Groups
                - Folders
                - Work Types
                - Formatters
                - ...
        - Custom UI

Implementation teams should follow standard development practices for check-in\check-out, commenting and labelling sources.

Solution artefacts should be managed individually under source control to best facilitate team based working and artefact change control.

14.2.   QA Testing

The test environment deployments should be established directly from labelled releases only.

The test environment should always be rolled back to its original state prior to the deployment of a release to ensure the installation is reflective of the installation into the target environment.

For example, if a customer has an environment at v2.1, then all internal QA releases prior to the shipping of v2.2 should all be applied against the same v2.1 environment, i.e. subsequent internal releases should not be applied on top of previous internal releases.

In addition to the option of snapshotting a VM, the databases can be backed-up and restored as a means of reestablishing the original KTA state.

15. Release Management

All releases should be:

- derived from a labelled source
- accompanied by release notes
- accompanied by installation notes

The release media:

- should use a consistent folder configuration for the deployment components. A suggested structure may include:

  - Release x.x.x.x
    - Documentation
      - [Release Note]
      - [Install Guide]
    - Database
      - Master Scripts
      - Upgrade Scripts
    - KTA
      - [Package(s)]
    - Custom UI

- can be maintained under source control (however a network location should suffice).

16. Performance Testing

It is imperative to consider system performance when designing and implementing a KTA solution.

The target performance should be known from the outset, and testing at each release should include a performance testing phase to ensure performance has been achieved and\or maintained with subsequent releases containing new or modified functionality.

Whilst an internal performance environment may not always reflect a target production environment, relative performance can be tracked through the lifetime of the project.

When establishing an environment for performance testing, aim to ensure the underlying database repositories are populated to reflect a realistic production environment (where feasible).

17. Component Startup/Shutdown Procedures

It is important to shutdown and startup the components of the TotalAgility platform in the correct order, to ensure that the system will start/stop successfully and take into account the dependencies of all components and ensure that users of the system are not impacted. This is particularly important where the components have been scaled across multiple servers or the platform supports a business critical function within the customer organisation.

Shutdown the components of the TotalAgility platform in the following order to ensure that the services shutdown correct:

1. TotalAgility Web Server
   a. IIS
   b. Kofax TotalAgility Streaming Service
2. Analytics Web Server
   a. IIS
3. Reporting Server(s)
   a. Kofax Reporting Service
4. Analytics Application Server
   a. Insight Bridge Service
   b. Insight Scheduler Service
5. TotalAgility Application Server
   a. Kofax Import Connector – Electronic Documents – Message Connector
   b. Kofax TotalAgility Core Worker Service
   c. Kofax TotalAgility Export Worker Server
   d. Kofax TotalAgility Streaming Service
   e. IIS
6. Transformation Server(s)
   a. Kofax Transformation Server
7. License Server(s)
   a. Kofax License Service
8. SQL Server

Startup the components of the TotalAgility platform in the following order:

1. SQL Server
2. License Server(s)
   a. Kofax License Service
3. Transformation Server(s)
   a. Kofax Transformation Server
4. TotalAgility Application Server
   a. IIS
   b. Kofax Import Connector – Electronic Documents – Message Connector
   c. Kofax TotalAgility Core Worker Service
   d. Kofax TotalAgility Export Worker Server
   e. Kofax TotalAgility Streaming Service
5. Analytics Application Server
   a. Insight Bridge Service
   b. Insight Scheduler Service
6. Reporting Server(s)
   a. Kofax Reporting Service
7. Analytics Web Server
   a. IIS
8. TotalAgility Web Server
   a. Kofax TotalAgility Streaming Service
   b. IIS

18. Programming against the TotalAgility SDK

KTA exposes the SDK as:

- A .Net component
- A SOAP webservice
- A HTTP Service

When executing a KTA method from within a KTA form or KTA Process, the SDK .NET component should be used, as this is the quickest to configure and the most efficient execution path.

When executing a KTA method from outside KTA (e.g. custom UI or Integration component), the SDK .NET component may be used, however given that this component (in this out-of-process context) will execute KTA web service calls anyway it can often be easier to configure a direct web service call (e.g. visual studio automatically generates the web service configuration) than to manually apply a configuration and ensure that supporting dlls are present within the project references.

The HTTP service is available (and is used internally by KTA forms). However, since automated parameter mapping to these API methods (e.g. JSON in and out) is not available in KTA, this is a more time consuming interface to program against compared to the use of the available SOAP approach.

19. Housekeeping and System Maintenance

19.1.   Maintaining Finished Jobs and the Audit Log

Ensure the customer's requirements are understood with regard to retention of process history and audit information.

Based upon the required retention period, put in place a suitable strategy to purge historical data, rather than letting the database grow indefinitely to the detriment of performance and\or storage space.

KTA supports a variety of approaches – the customer should adopt an approach appropriate to their business and IT environment. Options include:

- Manual – Administrator housekeeping can include manual archive of finished jobs\audit information on a regular basis using the KTA Workspace 'Purge' UI.
- Utilise the KTA Retention Policy Configuration settings for
  - Audit History
  - Finished Jobs
  - Documents
- Configure a KTA Scheduled Job which invokes the purge functionality exposed by the KTA SDK. The process can be configured to execute on a regular basis and purge records older than the required retention period.

19.2.   Document Storage and Retention

Ensure the customer's requirements are understood with regard to retention of documents in KTA.

Based upon the required retention period, put in place a suitable strategy to purge historical documents.

It is important to note that KTA is not a content management platform. The KTA document storage is intended to be a transient storage for documents during processing only. Upon completion of a case/process instance, associated documents should be transferred to the appropriate system of record, such as Perceptive Content.

If the documents database is allowed to grow indefinitely, this will be to the detriment of performance and\or storage space.

20.1. **Objectives**

The primary objectives when performing a production upgrade are to:

- Minimise Downtime
  - Minimise the amount of time the system is offline.
- Ensure a successful upgrade
  - Use a proven approach in a controlled and consistent fashion

To that end, one of a no. of upgrade strategies can be adopted (and adapted) to meet the needs of a specific environment. Three upgrade strategies are discussed in this section.

It is expected that the chosen approach will be determined by a combination of factors incl.

- Scale of Deployment
- Availability Requirements
- Criticality of the Solution
- Infrastructure capability

Ensure that you read the installation and upgrade guide and Kofax software compatibility matrix carefully before upgrading. Some components that are compatible with prior releases (such as operating system, database, .NET Framework or related product versions) may not be compatible with new releases and may require upgrading as part of the upgrade process.

Note: It is important to backup your environment prior to upgrading, particularly the TotalAgility databases but also the servers if possible. This minimises the risk of any issues occurring during the upgrade, as the servers and configuration can be easily restored, the problem rectified, and the upgrade re-attempted.

20.2. **Strategy 1 - Take Offline and Upgrade**

20.2.1. **Approach**

This approach involves:

- Taking all services offline
- Upgrading each server
  - Upgrading the dependent software or operating system, if required
  - Upgrading the database with first KTA server upgrade
- Bringing each server online

Note: Depending on the length of the upgrade window, it may be desirable to bring services online in priority order to re-enable partial service as quickly as possible.

20.2.2. **Verification**

This approach requires that prior to production upgrade:

- An upgrade has been performed in a lower order environment using the exact same process that will be applied to production.

- An upgrade has been performed in another environment using a copy of the production database.
- The upgrade has been successfully verified.

20.2.2.1. **Production Upgrade Preparation**

This approach does not require any specific preparation.

20.2.2.2. **Production Upgrade Test**

The production test comprises executing the following steps in a non-production environment.

- Stop All Services
- Upgrade the dependent software or operating system, if required
- Upgrade the first KTA server (and select to upgrade the database)
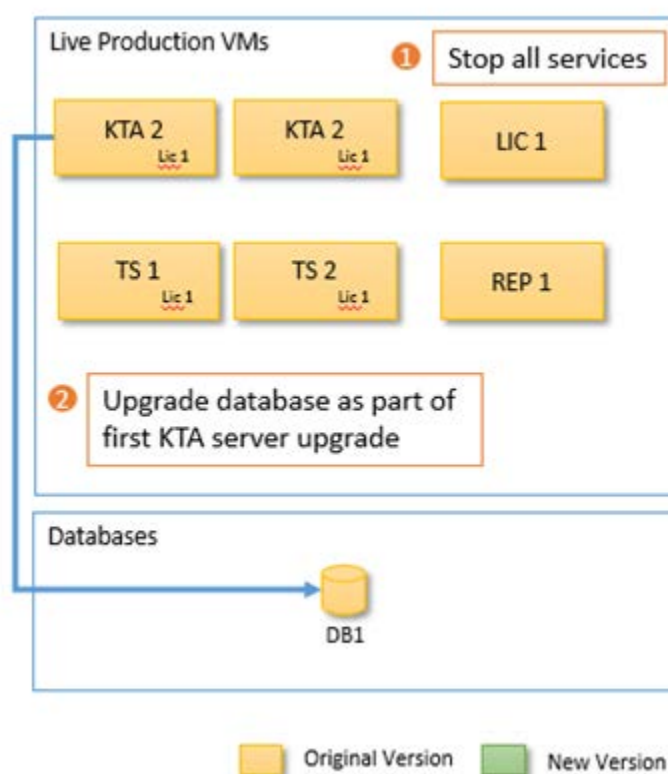
Figure 1 Upgrade Production Database

- Upgrade the license server.
- Start the first KTA server and perform initial verification.
- Upgrade a Transformation Server and perform initial verification.
- Upon successful verification, upgrade the remaining services
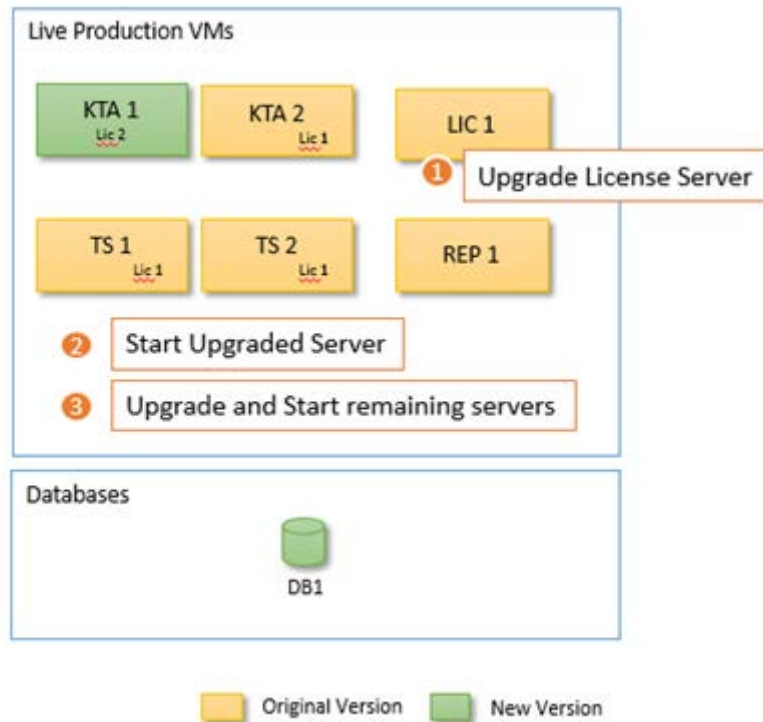
Figure 2 Upgrade License Server and Remaining Servers

- Verify a successful upgrade.
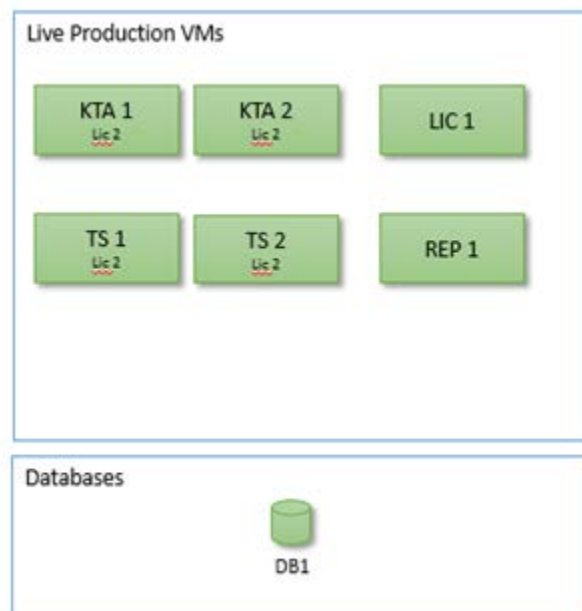
Figure 3 Production Upgrade - Final State

20.2.2.3. **Production Upgrade**

This strategy involves undertaking the same process in the production environment as per the test environment.

20.2.3. **Pros**

- Does not require any additional hardware \ capacity (even temporarily) due to in-place upgrades.
- May be acceptable where:
    - Downtime represents a reasonable operating window (e.g. overnight or at weekend), or
    - Additional capacity already exists, or reduced capacity is acceptable during upgrade.

20.2.4. **Cons**

- Represents the largest downtime as no pre-offline upgrades have been carried out.
- May be unacceptable, where downtime must be minimised e.g. 24\7 operation.
- Doesn't facilitate validation of the actual target instances prior to the production upgrade.
    - Approach relies upon the controlled and consistent execution of a pre- tested process.

20.3. **Strategy 2 - New Servers, Database Upgrade and Switch-Over**

20.3.1. **Approach**

This approach is characterised by the provisioning of a new set of servers that mirror production.

Upgrade comprises:

1. Installing new servers at the target version (against a new database)
2. Validating the installation
3. Taking the new services offline
4. Reconfiguring the new services to point at production database
5. Taking the production system offline
6. Upgrading the database
7. Starting the New Services
8. Verifying the new services
9. Decommissioning the old servers

20.3.2. **Verification**

This approach requires that prior to production upgrade,

- The new services have been verified against the newly installed databases (optional).
- The new services have been verified against a copy of the production database.

20.3.2.1. **Production Upgrade Preparation**

Install and validate the new server instances (against the new database)
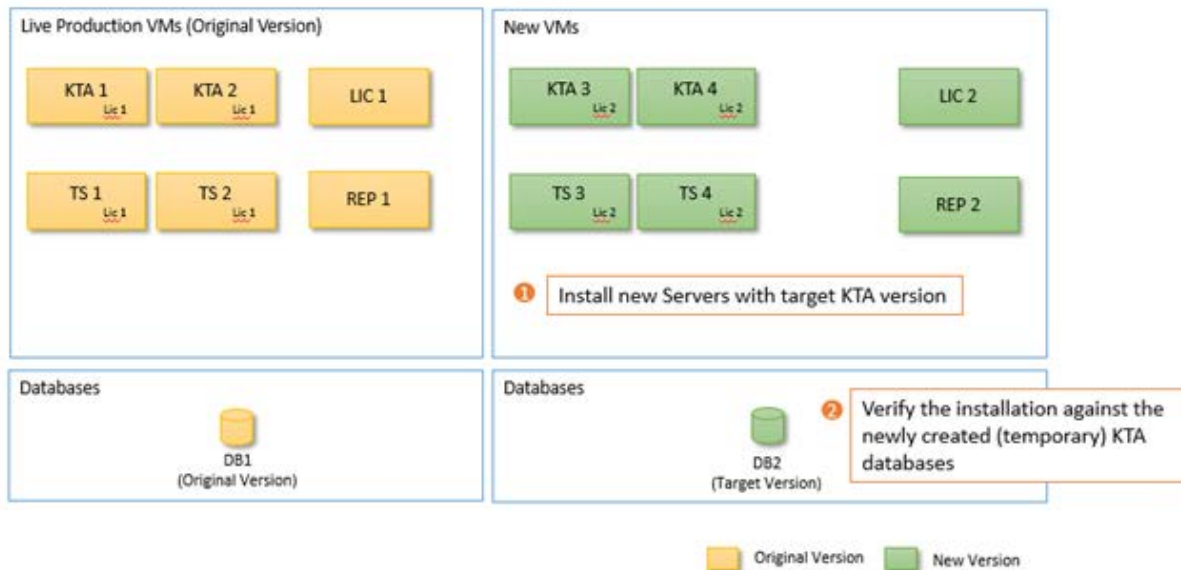
Figure 4 [1-2] Installing new servers at the target version (against a new database)

20.3.2.2. **Production Upgrade Test**

- Upgrade a copy of the Production DB.
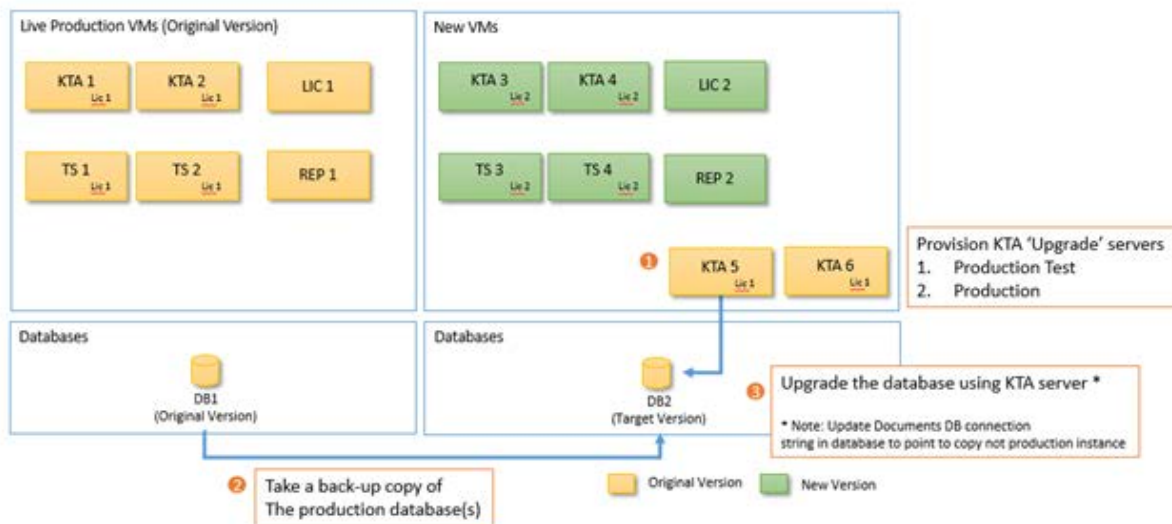
Figure 5 Upgrade Copy of Production DB

- Verify the upgrade by re-activating the new services.
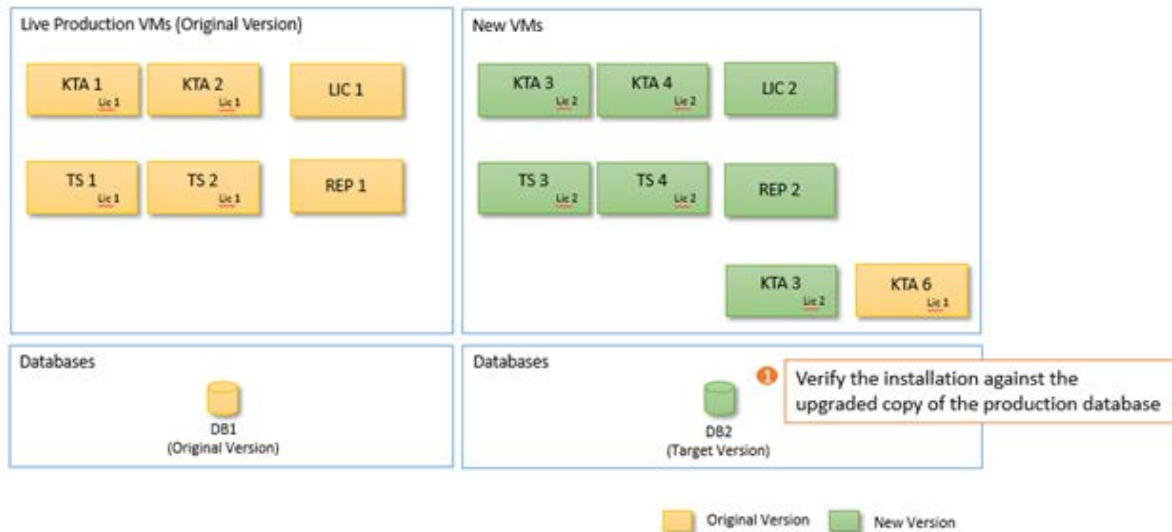
Figure 6 Verify Updated Copy of Production DB

20.3.2.3. Production Upgrade

- Update the new services to point at the production database
- Take Production offline.
- Upgrade the production database
- Start the new services
- Verify the installation
- Decommission the old production servers (and the 'upgrade' servers)
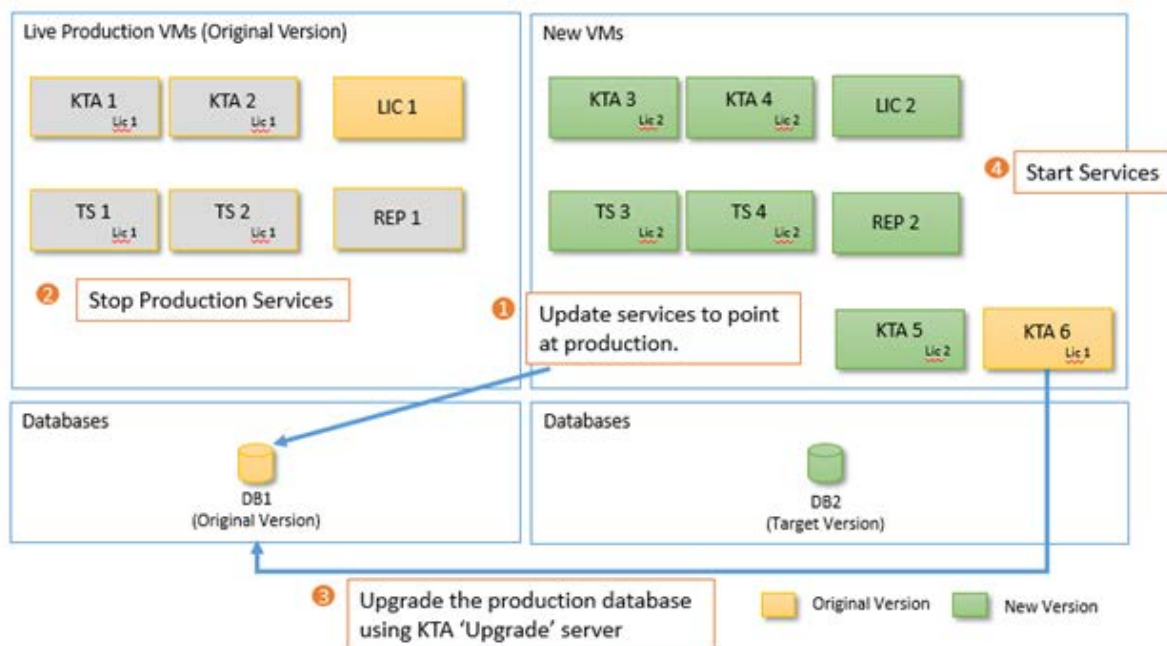
Figure 7 Production Database Upgrade
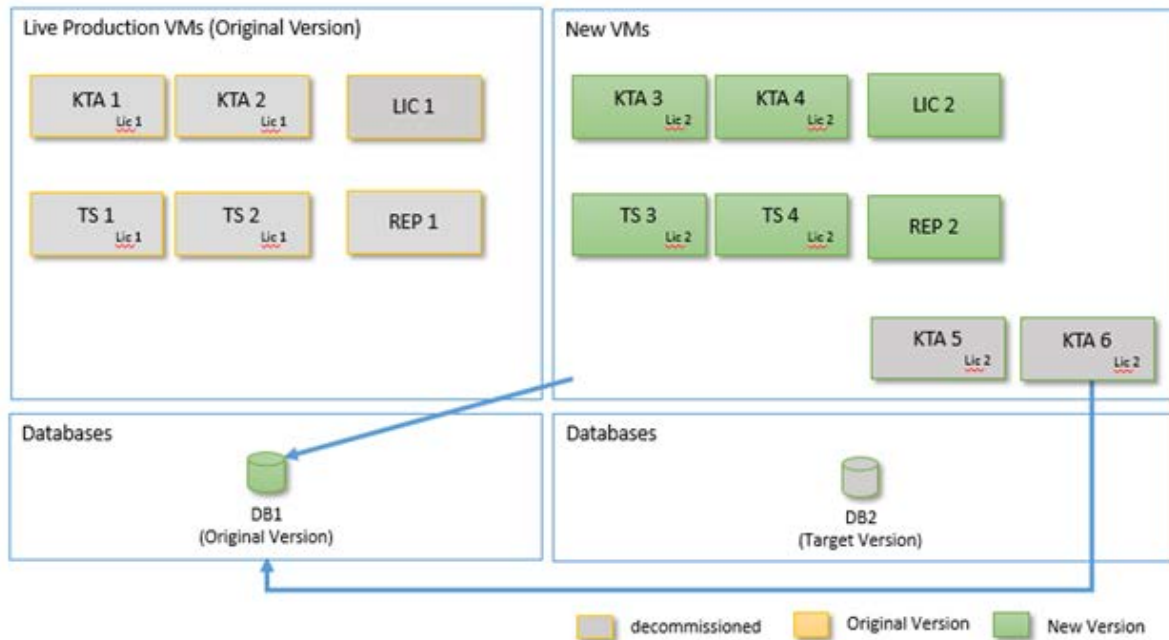
Figure 8 Production - Final State (incl. decommissioned services)

20.3.3. Pros

- Minimal downtime as system is only offline for database upgrade.
- Full capacity is restored immediately upon database upgrade.
- Facilitates clean installs on new servers (which may be preferable).
- Enables Validation of the target instances prior to production upgrade
  - against temporary databases
  - against copy of production
- Any upgrade issues with the server instances will not impact production.

20.3.4. Cons

- Requires capacity to resource duplicate hardware\VM resources matching production. The larger the deployment the larger the issue.

20.3.5. Comments

- A copy of the production database can be used to validate the installation and the database upgrade prior to the final production (database) upgrade.
- Note: Requires a KTA Server at the current version to orchestrate the database upgrade to the target version.
- A DR Site may represent a suitable environment to host production whilst new installs\upgrades are carried out at the primary site.
- The DR site would then be upgraded once the system is switched back to the primary site.

20.4. Strategy 3 – Some New Servers & Some Upgrades

20.4.1. Approach

This approach is a combination of Strategies 1 & 2, characterised by:

- the provisioning of some new servers at the target version
- upgrading some existing servers

Factors influencing the decision to install new vs. upgrade a given instance may include:

- Installation complexity
  - e.g. no. of separate installs to reach the target version
- Desire to validate the installation off the upgrade critical path.
- Solution profile\Service instances\Capacity
  - May plan to:
    - pre-provision a skeleton service that can be activated immediately upon database upgrade.
    - Bring additional services online as and when they are upgraded.

20.4.2. Verification

- For services targeted for upgrade, this approach requires that prior to production upgrade:
  - An upgrade has been performed in a lower order environment using the exact same process that will be applied to production.
  - An upgrade has been performed in another environment using a copy of the production database.
  - Any dependent software or operating system has been upgraded, if required.
  - The upgrade has been successfully verified.

- For services targeted for new install, this approach requires that prior to production upgrade:
  - The new services have been verified against the newly installed databases (optional).
  - The new services have been verified against a copy of the production database.

20.4.3. Pros

- Facilitates:
  - Clean installs on new servers where preferred
  - Offline validation of some installations against temporary databases (incl. copy of production)
  - Requires less spare capacity to resource duplicate hardware\VM resources than Strategy 2.

20.4.4. Cons

- Requires some capacity to resource duplicate required hardware\VM resources. The larger the deployment the larger the issue.

21. Monitoring and Resilience

Kofax Monitor for TotalAgility is an available add-on to a KTA environment. It provides a means of performing system level monitoring, for example of required Windows services, and supports stronger resilience in a KTA solution.

Common monitoring requirements include:

- Monitoring Windows Servers on each of machine to assess proper running, and alert and restart if necessary;
- Monitoring of events raised to the Windows Event log, for any warnings or errors, adding alerts and consolidating into a list for display to an administrator user from a single centralised console;
- Monitoring of each of the web server/web service end-points, for Web Server, App Server, License Server etc.;
- Testing the integration end-points for each external integration components, e.g. SharePoint, CRM, TRIM etc.

21.1. Windows Services

The following Windows Services should be monitored on each of the servers within the TotalAgility Platform, to ensure that the services are running, and to restart if the services fail.

Server	Services
Web/App Server	World Wide Web Publishing Service
	Kofax Import Connector - Electronic Documents - Message Connector (where used)
	Kofax TotalAgility Core Worker
	Kofax TotalAgility Export Worker (where used)
	Kofax TotalAgility Streaming Service
Transformation Server	Kofax Transformation Server
License Server	Kofax License Server
Reporting Server	Kofax TotalAgility Reporting Server

Analytics Server	World Wide Web Publishing Service Insight Bridge Service 5.2.0 Insight Scheduler Service 5.2.0
Kofax Search and Matching Server	Kofax Search and Matching Server Kofax Search and Matching LoadBalancer (where applicable)

21.2. Web Server End-Points

The following web server end-points should be monitored, to ensure that they return a response. If the services are not returning a response, it may mean that a system error has occurred, or IIS has failed, requiring further investigation by a system administrator to resolve.

Server	Endpoint URL
Web Server	http://[Web Server FQDN]/TotalAgility/Designer http://[Web Server FQDN]/TotalAgility/forms http://[Web Server FQDN]/TotalAgility/Services/SDK /ServerService.svc
Application Server	http://[App Server FQDN]:/TotalAgility/Services/Core /ServerService.svc
Transformation Server	N/A
License Server	N/A
Reporting Server	N/A
Analytics Server	http://[Analytics Server FQDN]/Insight/Admin http://[Analytics Server FQDN]/Insight/DataLoad

	http://[Analytics Server FQDN]/Insight/reporthtml
	http://[Analytics Server FQDN]/Insight/Studio
	http://[Analytics Server FQDN]/Insight/Themes
	http://[Analytics Server FQDN]/Insight/View
Kofax Search and Matching Server	N/A

21.3. Windows Event Logs

The following events should be monitored, to detect errors within the Windows Services. This will include system errors as well as application errors. In some cases these errors are application related and may not require intervention from a system administrator. In the case of system errors, e.g. related system being unavailable, these errors will need to be resolved by a system administrator.

Server	Event Details
Web Server	**Event Log Name:** Application **Event ID:** 0 **Event Source:** TotalAgility Streaming Service **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
Application Server	**Event Log Name:** Application **Event ID:** 0 **Event Source:** TotalAgility **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
	Event Log Name: Application **Event ID:** 0 **Event Source:** TotalAgility Core Worker **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
	Event Log Name: Application **Event ID:** 0 **Event Source:** TotalAgility Export Worker **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**

	Event Log Name: Application **Event ID:** 0 **Event Source:** TotalAgility Streaming Service **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
Transformation Server	**Event Log Name:** Application **Event ID:** 0 **Event Source:** KofaxTransformationServerService **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
License Server	**Event Log Name:** Application **Event ID:** 0 **Event Source:** KOFAX-SAL **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
	Event Log Name: Application **Event ID:** 0 **Event Source:** KSALicenseService **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
Reporting Server	**Event Log Name:** Application **Event ID:** 0 **Event Source:** KofaxTAReportingServerService **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**
Analytics Server	**N/A**
Kofax Search and Matching Server	**Event Log Name:** Application **Event ID:** 0 **Event Source:** Kofax Search and Matching Server **Event Category:** Any **Event Level:** Critical/Error/Warning **Event Description:**

21.4. **Text Logs**

The text logs being used on each of the servers in the TotalAgility stack are as follows:

Server	Logs
Web Server	[Install Folder]\Kofax\TotalAgility\Agility.Server.Web\ OpenAPISDK1.log
Application Server	C:\ProgramData\Kofax\KIC-ED\MC\Log*.log [Install Folder]\Kofax\TotalAgility\Agility.Server.Web\ OpenAPISDK1.log
Transformation Server	[Install Folder]\Kofax\TotalAgility\Transformation Server\ KofaxCPUServerLog.log
License Server	C:\ProgramData\Kofax\KSALic\logs*.txt
Reporting Server	C:\ProgramData\Kofax\TotalAgility\Reporting\Log*.log
Analytics Server	C:\Temp\Insight_<version>*.log **Note:** There are multiple configuration within Insight where the log file location is configured, at least one for each of the sub-components.
Kofax Analytics for TotalAgility	C:\Temp\Insight_<version>\Consolidated.log **Note:** It is recommended to change this when installing KAFTA, to the location where the KAFTA executables have been deployed, i.e. C:\Program Files\Kofax\Insight <version>\KAFTA. This can be changed in the Consolidated.exe.config file in the KAFTA executables folder.
Kofax Search and Matching Server	Defaults to C:\ProgramData\Kofax\KSMS\Logs. Configurable during installation (and subsequently via the windows registry).

22.1. **TotalAgility Installation Prerequisites**

This section provides instructions for applying prerequisites to a target server for TotalAgility installation. This is split between Windows Roles and Features, and third-party software installation. These steps are based on Windows Server 2012 (R2).

Please refer also to the section above on "Installation Best Practice".

22.1.1. **Windows Roles and Features**

1. Login to the target server.
2. Open Server Manager.
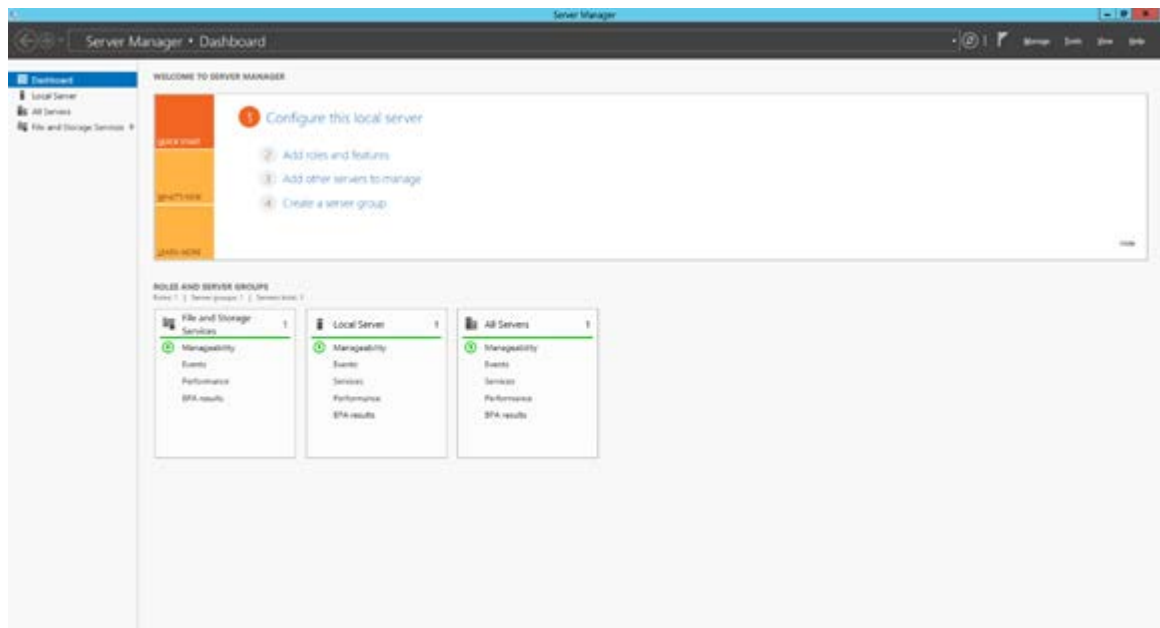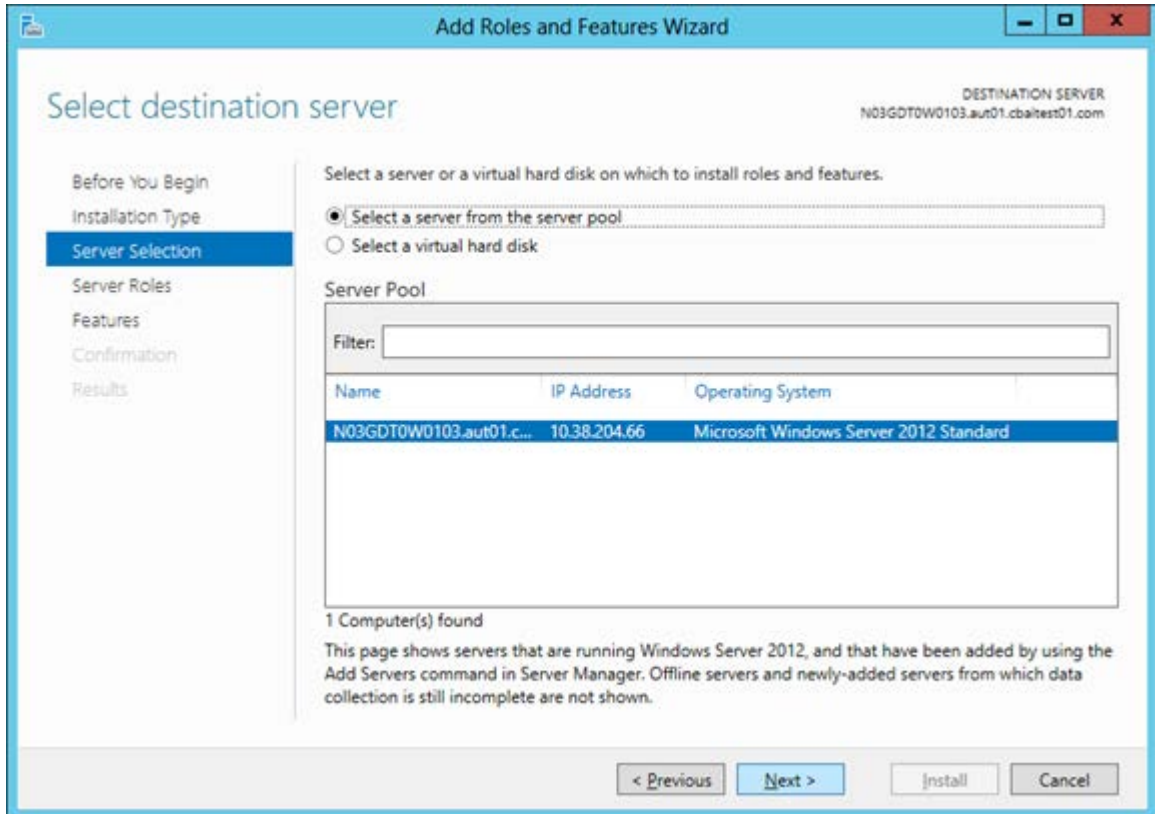
3. Select Add roles and features.
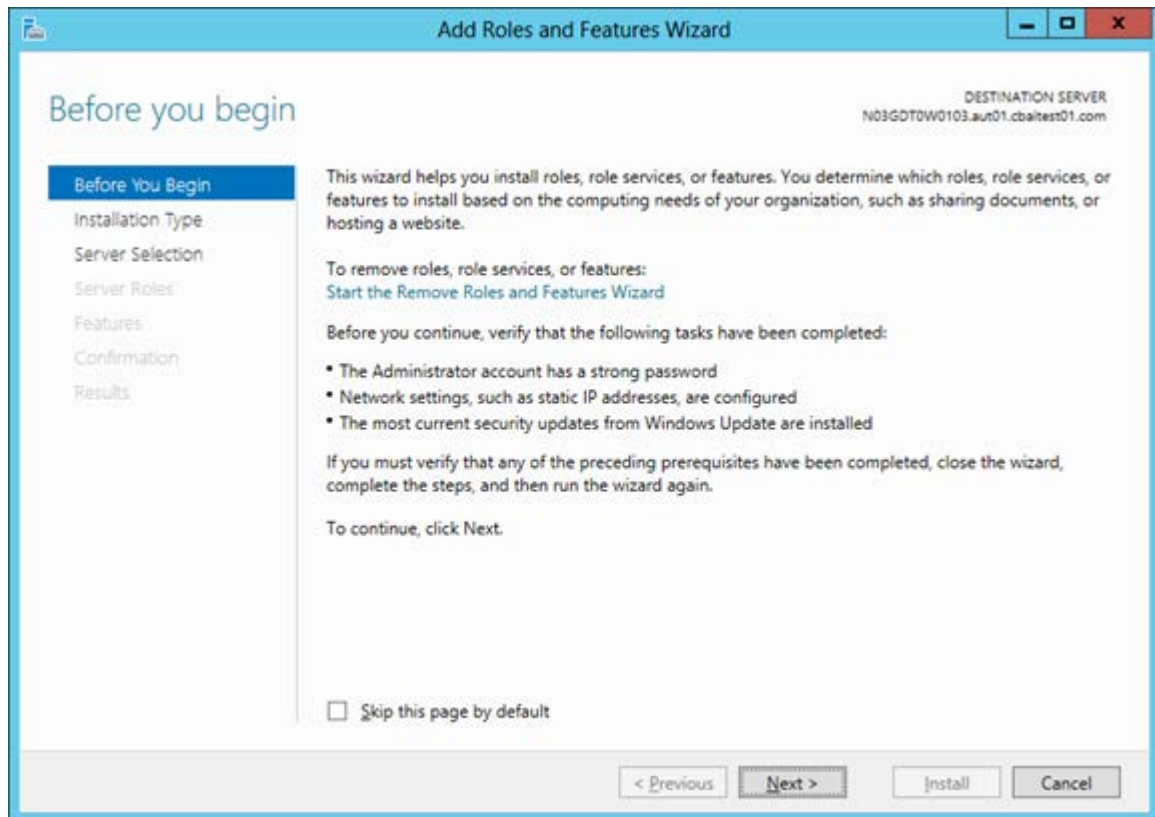
4. Click Next.
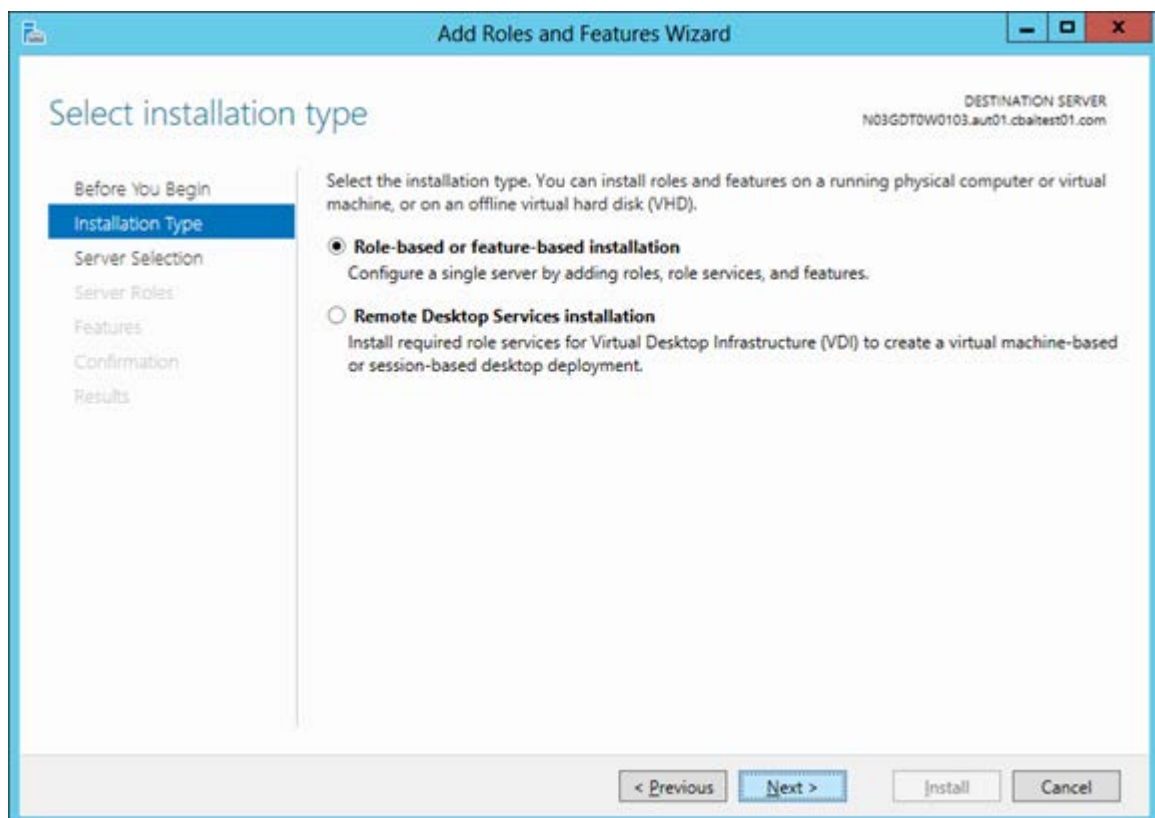
5. Select "Role-based on feature-based installation" and click Next.

6. Select the current server and click Next.
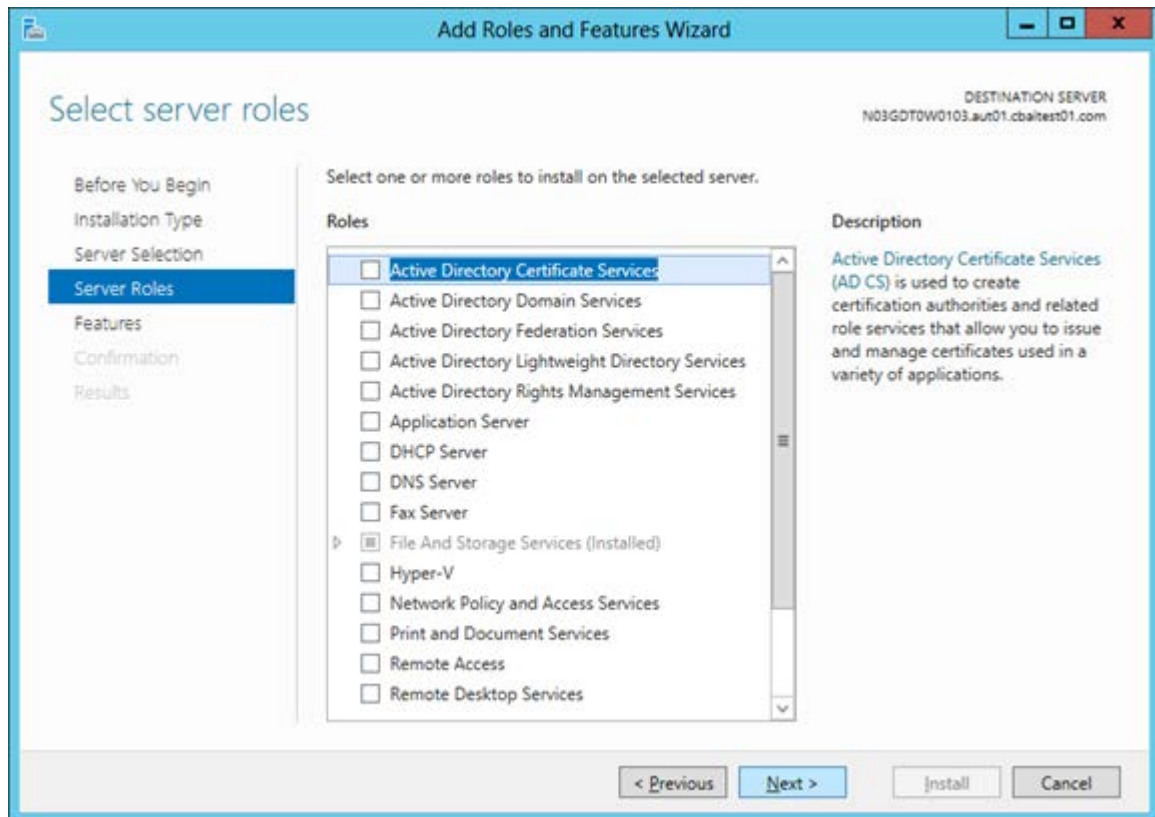
7. Add the Server Roles defined by the prerequisites listed for the target server type in the product Installation Guide, e.g. Web Server, Application Server.
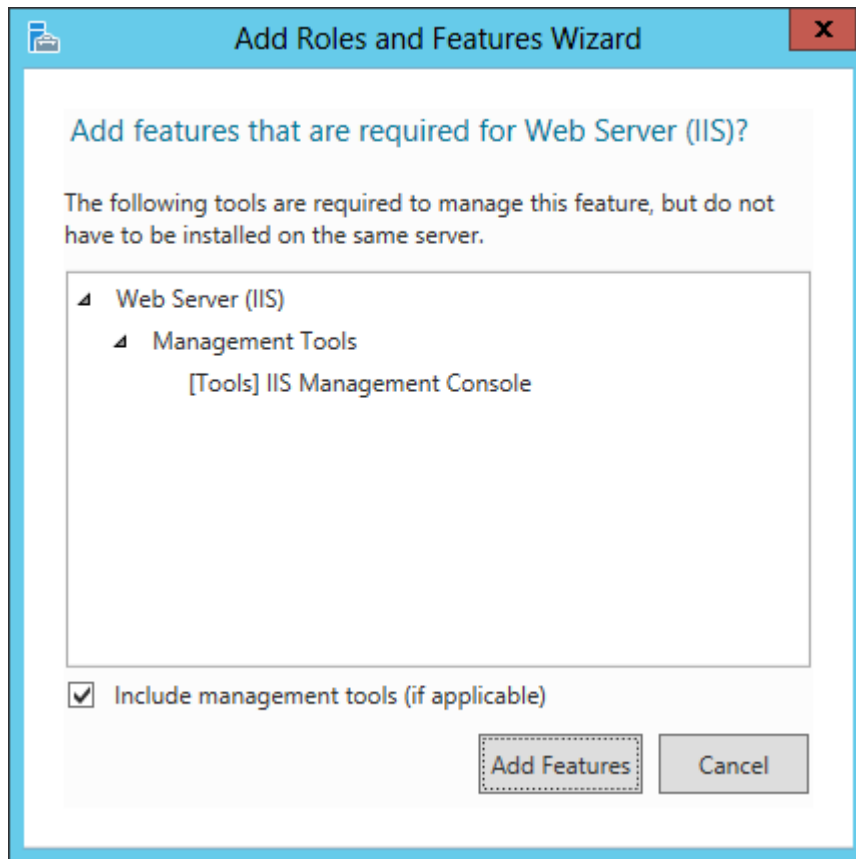
8. You may be prompted to select additional features required for the role, if so, click "Add Features".

9. Click Next to proceed to feature selection.
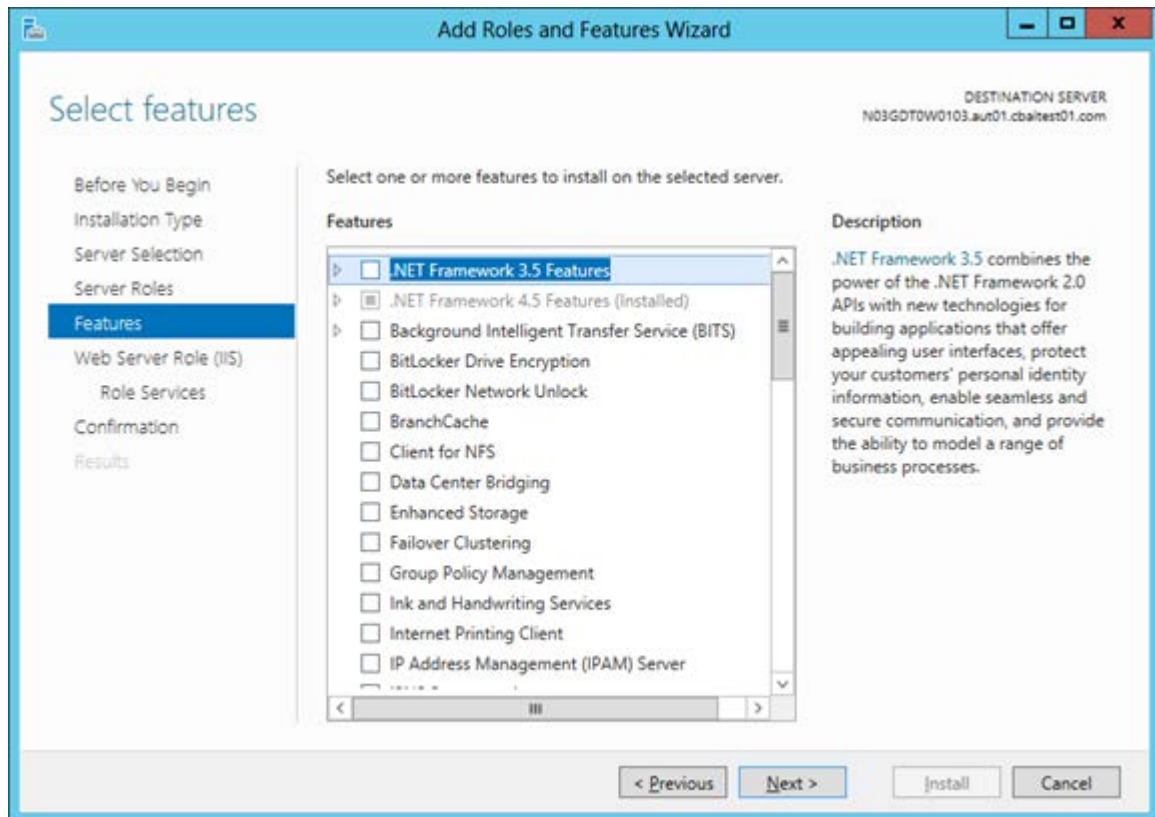
10. Add additional features for the roles selected as defined for the target server type in the product Installation Guide.
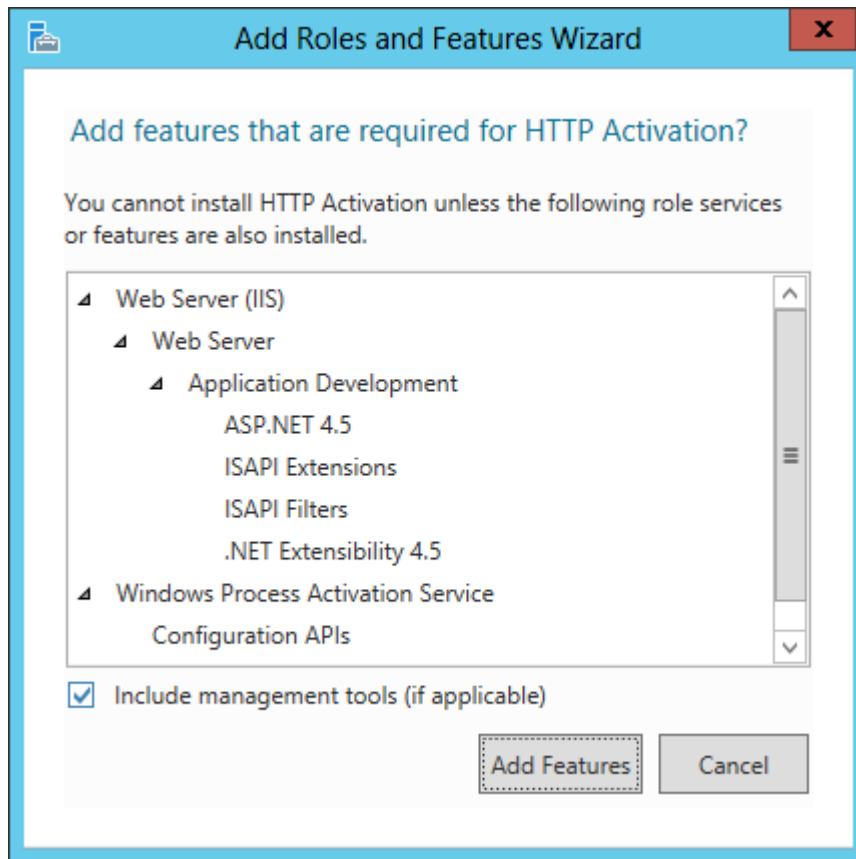
11. You may be prompted to add dependencies for certain features. If so click "Add Features".

12. Click Next.
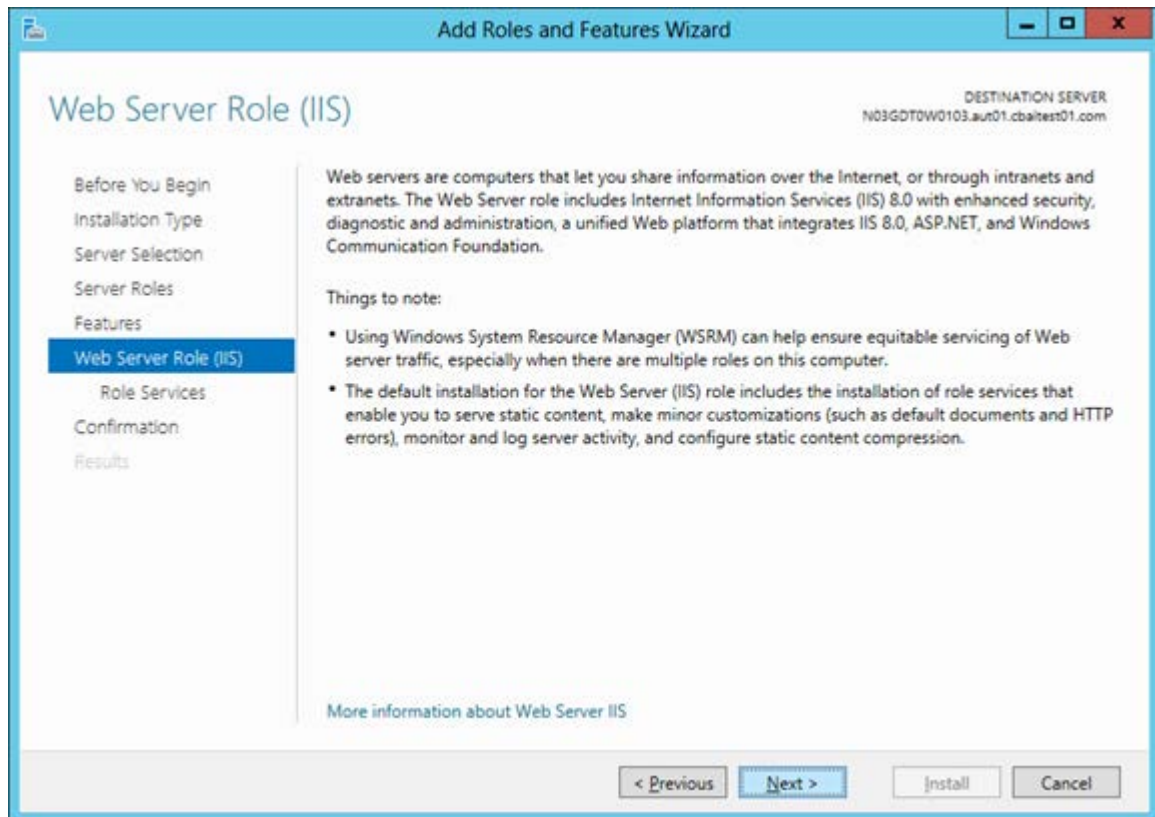
13. For "Web Server Role (IIS)" click Next.

14. For "Web Server Role (IIS)", select Web Server -> Security -> Basic Authentication and Windows Authentication as per the Installation Guide.
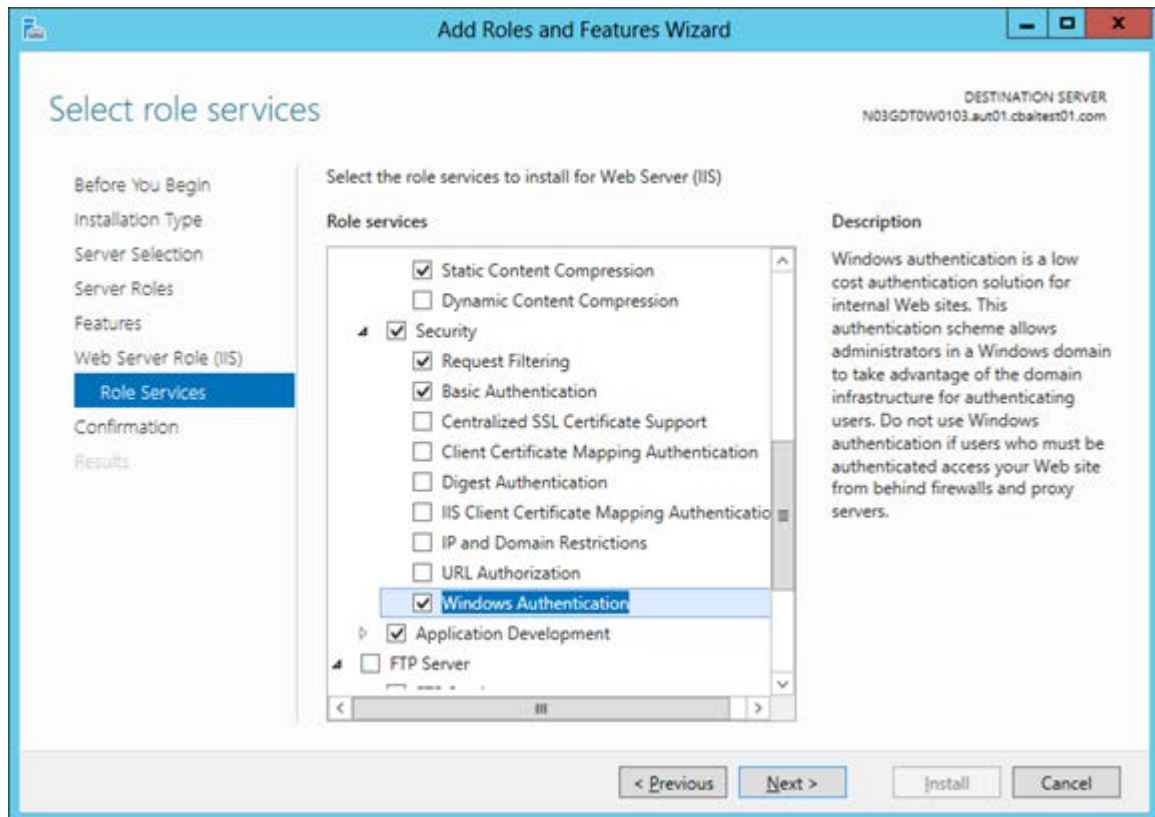
15. Click Next.
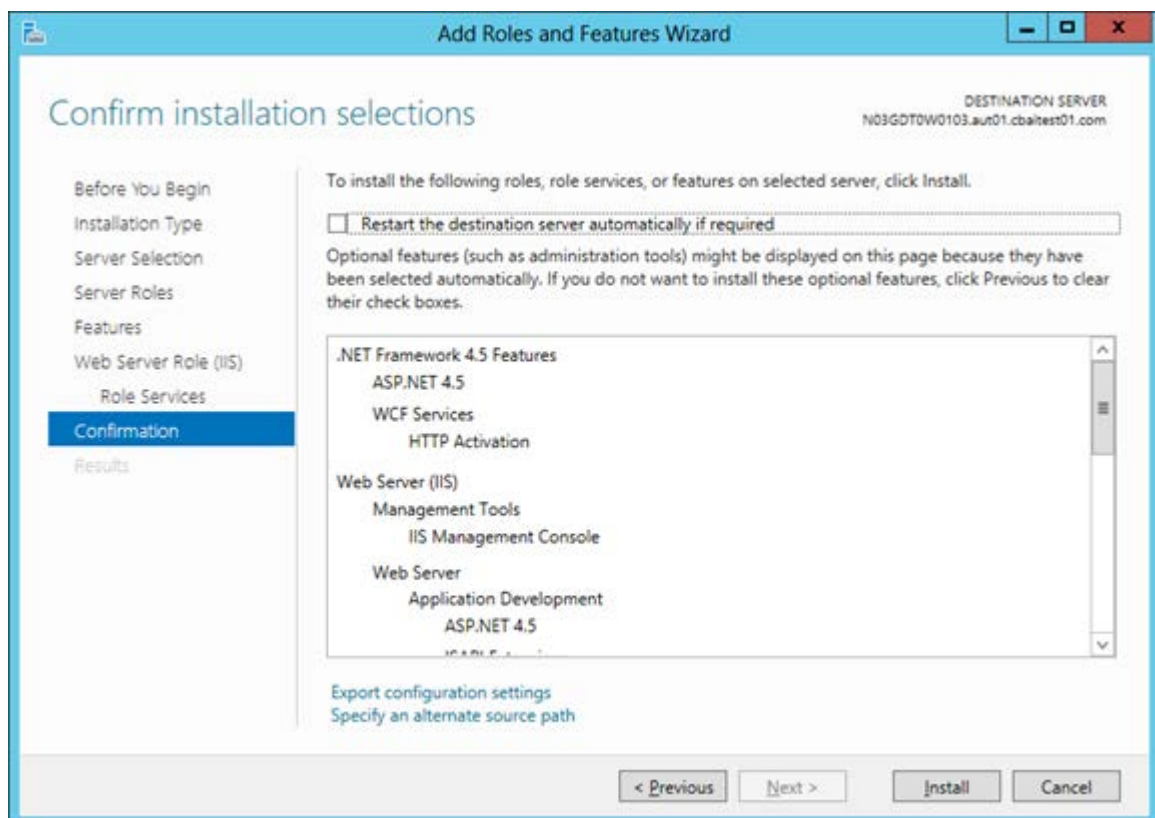
16. Click "Export configuration settings" to save the selected options as an XML file that can be applied to other servers of the same type in the same environment, or in higher environments.

17. Click Install.
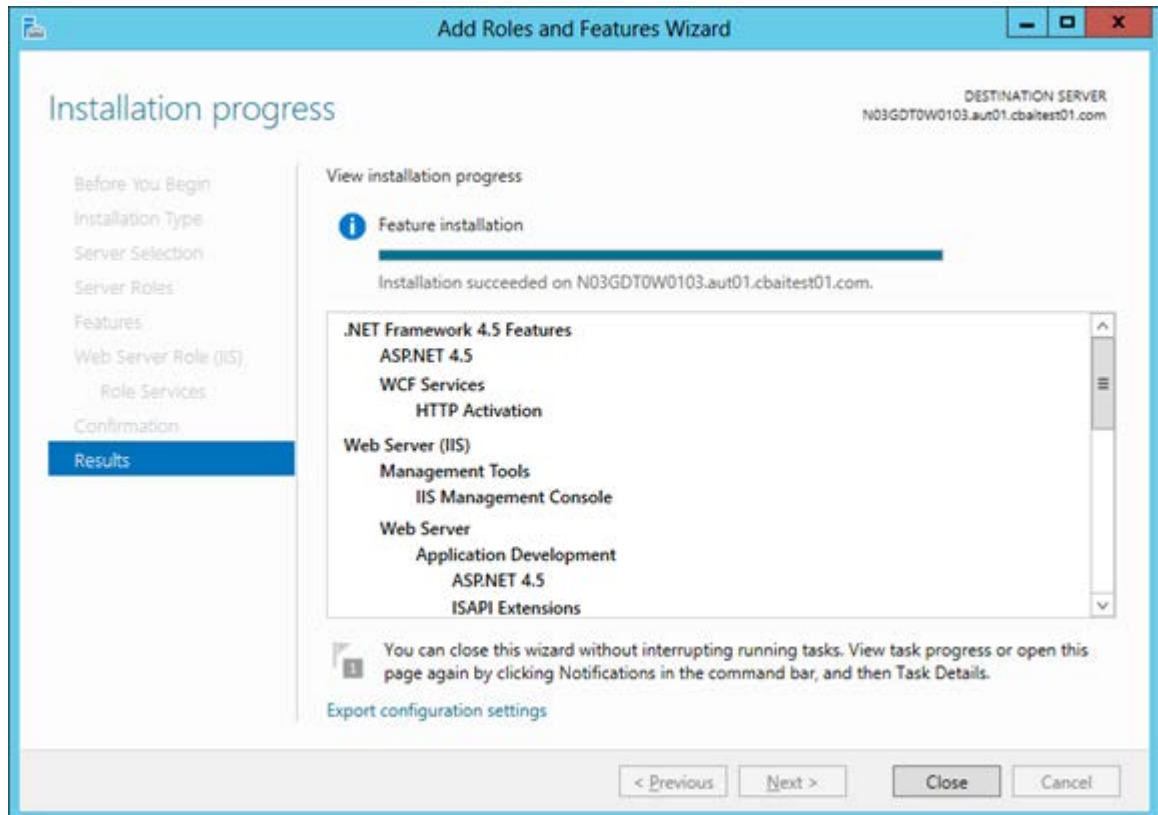
Click Close once installation has completed.

22.1.2. **Third-Party Tools**

The following command-line scripts can be used to silently install the third-party tools required on the TotalAgility Application Server using the MSI install files for each component. Each component needs to be downloaded from Microsoft and copied to the target server(s).

22.1.2.1. **Microsoft ODBC Driver 11 for SQL Server**

Install using the MSI silent installer script:

```
msiexec    /quiet    /passive    /qn    /log    msodbcsql.install.log    /i    msodbcsql.msi
IACCEPTMSODBCSQLLICENSETERMS=YES
```

22.1.2.2. **Microsoft Command Line Utilities 11 for SQL Server**

Install using the MSI silent installer script:

```
msiexec /quiet /passive /qn /log MsSqlCmdLnUtils.install.log /i MsSqlCmdLnUtils.msi
IACCEPTMSSQLCMDLNUTILSLICENSETERMS=YES
```

22.1.2.3. **SQL Server Native Client for SQL Server 2012**

Install using the MSI silent installer script:

```
msiexec    /quiet    /passive    /qn    /log    sqlncli.install.log    /i    sqlncli.msi
IACCEPTSQLNCLILICENSETERMS=YES
```

22.2. Pre-KTA7.3 pre-caching of Classification/Extraction projects on Transformation Server

In TotalAgility versions prior to 7.3, the classification and extraction projects are downloaded to the Transformation Server the first time they are used. This can lead to a delay the first time a document or folder using these projects are processed, which can present an issue for real-time processing from mobile or web applications.
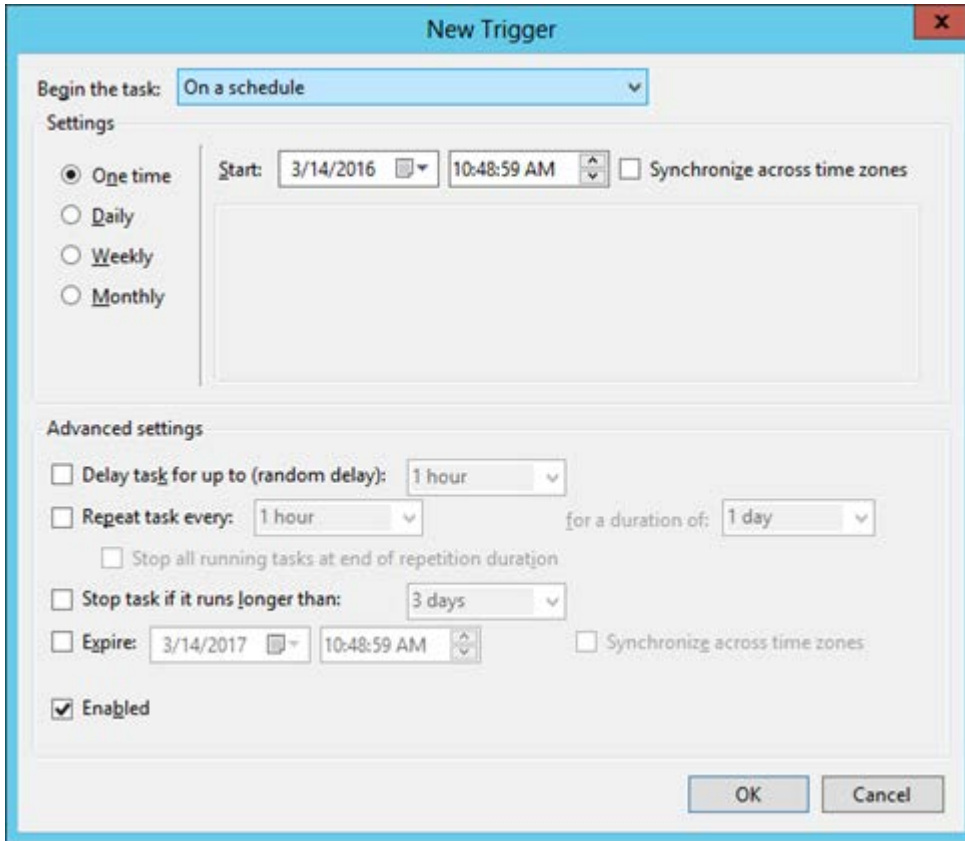
There is a workaround to pre-cache the classification and extraction projects, by creating a Windows executable, written in .NET, to call the Business Process web services and trigger a business process to perform classification and extraction for the target document class(s) using sample documents. This executable can be scheduled to run as a Windows Task, triggered when the Transformation Server service is restarted.

The trigger for the task is based on an event log trigger, see below for an example:

1. Click on the Triggers tab in the Windows Task.

2. Click New.

3. Select Begin the task -> On an event.
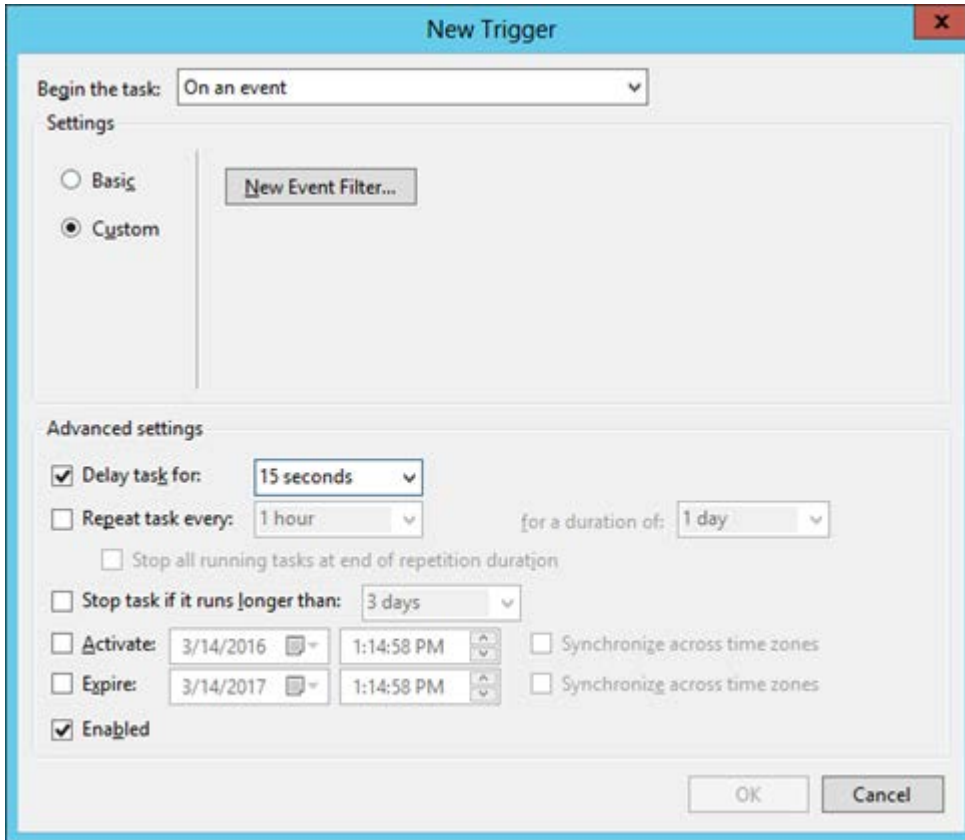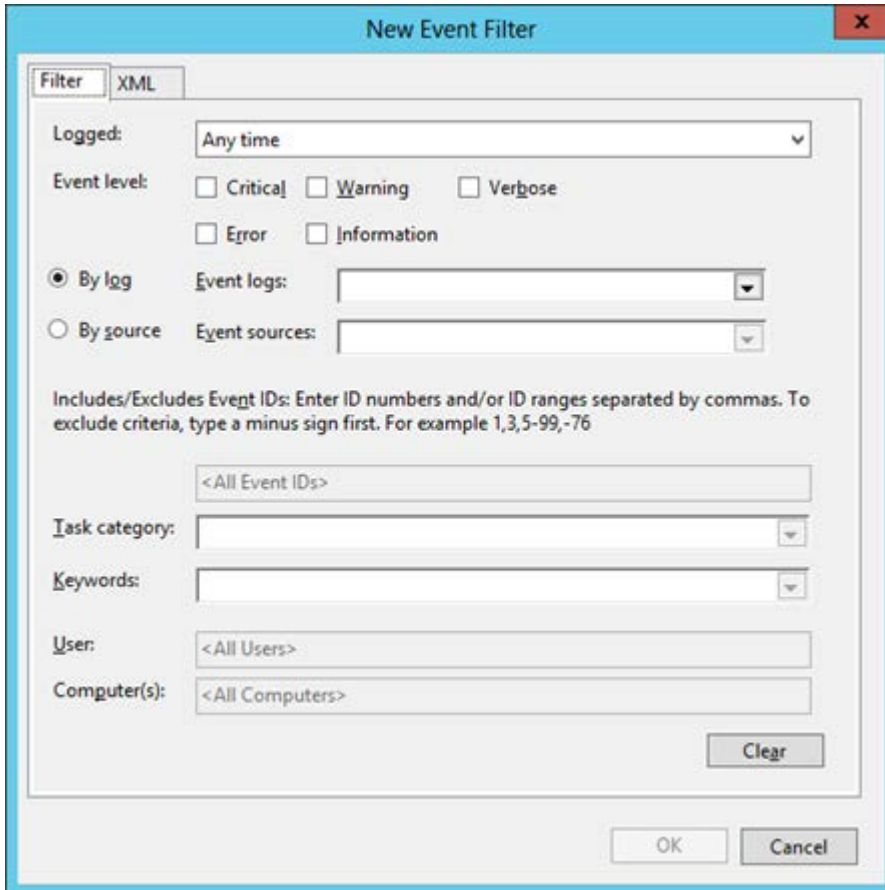
4. Select the Custom radio button and select "Delay task for" and enter "15 seconds" in the drop-down list.

5. Click "New Event Filter…".

6. Select:
   a. Event level: Information
   b. Event logs: Windows Logs -> Application
   c. Event sources: KofaxTransformationServerService

7. Click on the XML tab.

8. Click "Edit query manually".
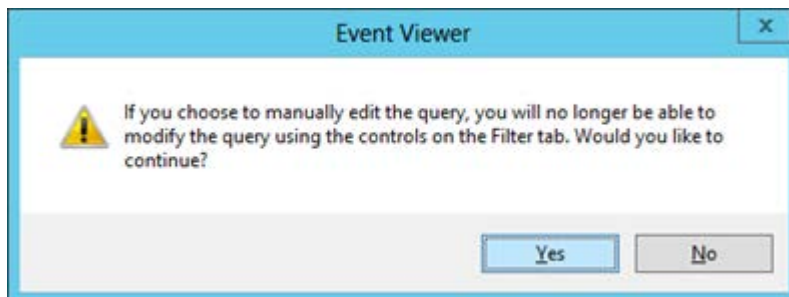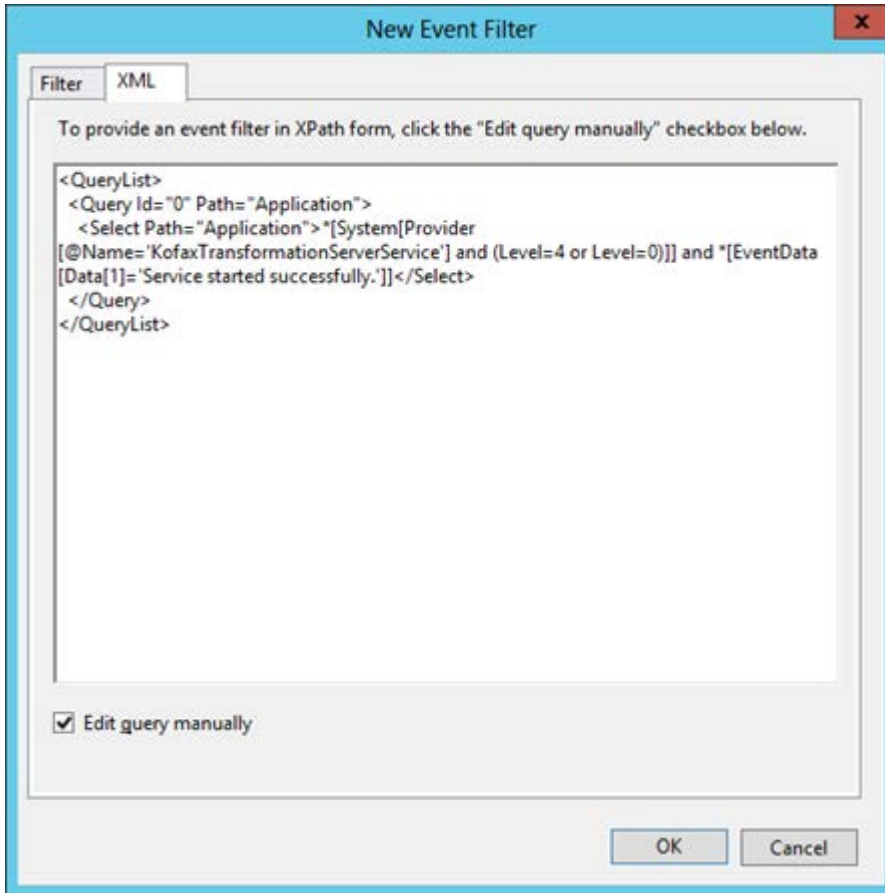
9. Click Yes.
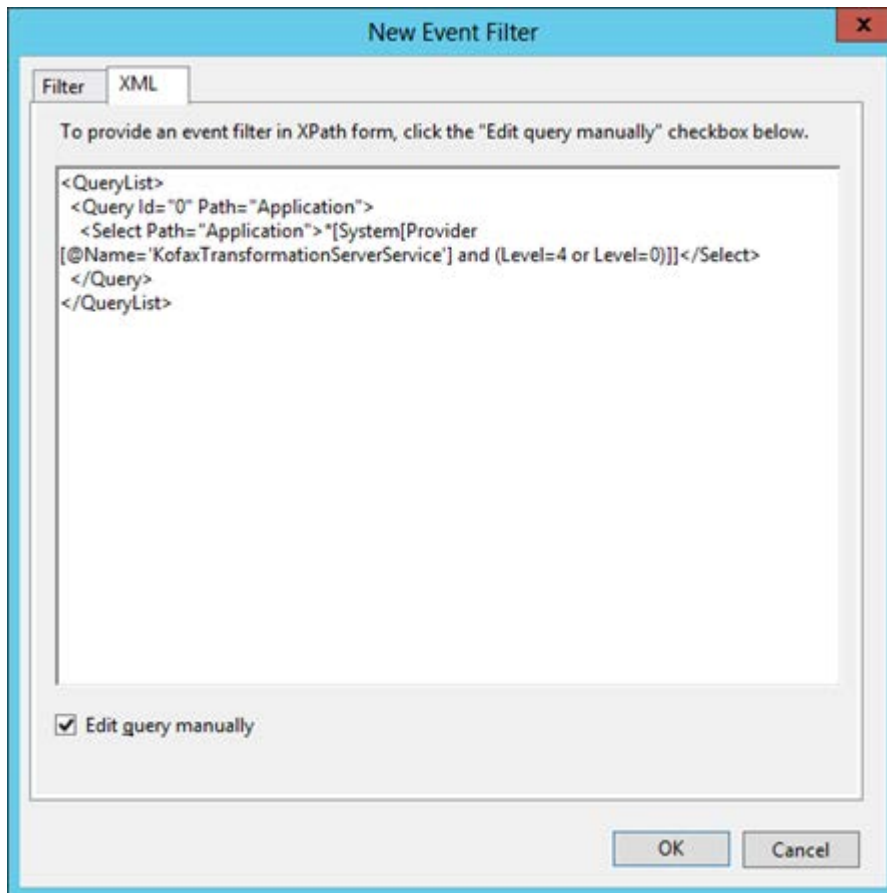
10. Change the query to the following:

```
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='KofaxTransformationServerService'] and (Level=4 or Level=0)]] and *[EventData[Data[1]='Service started successfully.']]</Select>
  </Query>
</QueryList>
```

11. Click OK.