opentext™

OpenText™ Intelligent Capture

System Overview

This guide provides an overview of the Intelligent Capture application. It describes the system design, requirements, and processes involved in capturing a document.

ECPCORE220200-GCS-EN-01

OpenText™ Intelligent Capture
System Overview

ECPCORE220200-GCS-EN-01
Rev.: 2022-Mar-15

This documentation has been created for OpenText™ Intelligent Capture CE 22.2.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Product Basics

This section provides a general description of Intelligent Capture.

1.1 Intelligent Capture Features

Intelligent Capture captures and processes documents from a variety of sources including scanners, fax servers, email servers, file systems, web services, and via RESTful web services. Document information can be stored as images, text, or both. Intelligent Capture is optimized for capturing documents, not storing them for long-term access. Typically, documents remain in the system for a few hours to a few days, until they are exported to a content repository or other back-end system.

Intelligent Capture is a scalable solution that optionally uses multiple servers to manage resources. Therefore, it can process large amounts of data from throughout your enterprise. It also handles multiple languages and system locale settings. Benefits of using Intelligent Capture include:

- Reducing operating costs caused by factors such as document preparation and data entry.
- Reducing recovery costs caused by mishandled physical documents.
- Improving information quality for critical business processes.
- Accelerating business processes by providing immediate access to all information and supporting documentation.
- Enforcing strong compliance control by storing documents and metadata electronically.
- Minimizing processing errors, improving data accuracy, and boosting productivity.

Note: The Intelligent Capture documentation assumes that you have access to all product features. Features requiring a special license are sometimes indicated in the documentation. However, if you require a specific feature, check with OpenText Global Technical Services at My Support (https://support.opentext.com) to determine licensing requirements. For licensing information, see *OpenText Intelligent Capture - Web Client Licensing Guide (ECPCORE-ARE)*.

1.2 The Information Capture Process

Intelligent Capture uses capture processes to convert information from printed documents, faxes, and email messages into digitized data, and to store the data and images into back-end systems for fast and efficient data retrieval. A process defines the modules that the Intelligent Capture Server uses to process images and data, the order in which to use those modules, and what to do with the resulting data. A typical process performs the following:

Captures information

- Captures paper, faxes, film, images, or imported electronic documents (structured and unstructured) through fax, scanner, network drives, remote sites, and via RESTful web services.
- Improves image quality, cleans up images to improve image clarity and readability, and speeds up processing without manual intervention.
- Enhances images to improve recognition results and organizes multi-page documents into document sets.

Classifies information

- Identifies documents so that they are routed to the appropriate data extraction processes.
- Enables operators to confirm or update document identification.

Extracts information

- Enables data extraction from identified documents.
- Performs optical or intelligent character recognition (*OCR*) to extract machine and handprint text using zonal OCR for structured documents and full-text OCR for unstructured documents.
- Reads bar codes to extract alphanumeric data.
- Enables key from image data processing.

Validates extracted information

- Maintains data integrity using restriction masks, regular expressions, and numeric only field properties.
- Validates data formulas against an external database or custom business rules using scripting events.
- Populates and validates data from an external source (for example, a database, Documentum repository, or text file).

- Enables operators to check, correct, and finalize extracted data.

Delivers information

- Exports both images and index data to leading content management systems, *ERP*, *BPM*, databases, and other systems.
- Supports conversion to *PDF*, full-text OCR, and PDF compression.

1.3 Sample Usage Scenario

This section describes a situation in which Intelligent Capture is used to simplify a business process.

Example 1-1: Processing a Loan Application

In this example, Intelligent Capture enables a bank to reduce costs by processing a loan application globally over the Internet. The process defines the steps used to capture the data in the faxed image and instructs the server on how to process the batch. The loan application is a structured document that contains data in the same area of every page, making it a good candidate for the extraction of data using an *OCR* module. The process includes the Web Services Input, NuanceOCR, Completion, and Web Services Output modules.

Starting the data capture process:

1. A customer completes a loan application at a local bank branch office in Stuart, *FL*.
2. A loan officer confirms the application is complete and faxes a copy to the credit department in the main office, which is located in Toledo, *OH*.
3. A bank employee creates a batch by sending the *URL* of the faxed image using a web service. For security, the bank uses *HTTPS* for all Internet transactions.
4. The Web Services Input module imports the image and creates a batch, storing the data in system variables known as IA values.

Capturing data:

1. The Web Services Input module sends the batch data to the Intelligent Capture Server located in Mexico City, Mexico. To reduce expenses, the bank uses the processing center in Mexico City to perform the extraction and indexing steps.
2. The server sends the task to the NuanceOCR module, which is configured to capture data from specific areas (or “zones”) on the page.

3. The customer's name, address, and social security number are extracted from the application and stored in IA values. The module passes the IA values to the server and the data is added to the batch.
4. The server sends the task to the Completion module for an operator to verify the data. The process defines index fields for the captured data, along with custom scripts that verify the address against a *US* postal ZIP Code database. The operator compares the data to the image and verifies that the captured data is accurate. The final data is then sent back to the server.
5. The server sends the extracted social security number to the Web Services Output module.
6. The Web Services Output module issues a web service call to a credit bureau requesting the customer's credit history.
7. The Web Services Input module receives the customer's credit history and passes the information to the Intelligent Capture Server.

Exporting data:

1. The server sends the task to the Web Services Output module.
2. The module exports the customer's credit card history and indexed fields to a back-end repository system located in Toledo, OH.

Using data:

1. A loan officer reviews the customer's credit history and approves the loan. The data remains on the back-end system for future use.
2. The bank credits the customer's account at the local bank in Stuart, FL.
3. A satisfied customer buys a boat.

1.4 Implementation and Production Basics

The following high-level steps are required to process data with Intelligent Capture:

1. Install the software, which consists of server-side components, design tools, administration tools, and production modules.
2. Set up your capture system environment, including system-wide configuration options.
3. Create and deploy *profiles* that specify processing options for images and define the user interface that operators see. Once deployed to the server, profiles can be used across multiple *capture processes*.
4. Create one or more *CaptureFlows* (graphics-based process models) that specify how batches are created and processed using attended and unattended steps.
5. Install an instance of the designed CaptureFlow on the test capture server. This step also compiles the process.

6. Upload all profiles, document types, and recognition projects that are required by the steps of the installed process to the test server. If the process uses the .NET Code module, upload the DLL with code as well.
7. Set up steps of the installed process. During setup, the process steps link the profiles, document types, recognition projects, and code. The modules use these resources in production.
8. Create a test batch based on the installed process, and use the production modules to process data in the batch. Some processing steps can run as an unattended service, others require operators to manually process the data.
9. When you are satisfied with the results of the test batch, deploy (upload) the service components created in Intelligent Capture Designer to the production capture server. The service components include profiles, document types, queries, styles, CaptureFlows, and other resources.

CaptureFlows can be simple or complex, and can include some supplementary tasks that might apply to your situation, including:

- Setting up departments, user roles, and permissions to distribute operator tasks and protect private data.
- Setting up third-party software to send or receive data from Intelligent Capture.
- Implementing customizations to perform tasks that are not provided as built-in features.

Chapter 2

System Design

This section describes the design of Intelligent Capture.

2.1 Intelligent Capture Architecture

The server is an open integration platform that manages and controls the document capture process by routing document pages and processing instructions to client modules, which are also called *production modules*.

Note: For historical reasons, the Intelligent Capture Server is often referred to as the InputAccel Server.

The production modules are software programs that perform specific information capture tasks such as scanning pages, enhancing images, and exporting data. In addition to the modules that ship with the product, Intelligent Capture supports third-party certified modules. Modules use *TCP/IP*, an industry-standard network protocol, to connect to the server. Two modules, Web Services Input and Web Services Output, can communicate over the Internet using *SOAP*. This enables Intelligent Capture to exchange data with other web services systems regardless of their locations, operating systems, or platforms.

Intelligent Capture processes data in collections called batches. Within each batch, pages are grouped and organized in a tree structure. Most production modules can process data at any level of the tree, as specified by the process. Sometimes you need a document, or even an entire batch, to be handled as a single unit. But many times it is sufficient for a module to process one page at a time, and this can speed up processing. Each task in a process is self-contained, so modules can process tasks from any batch in any order. The server tracks each task in a batch and saves the data generated during each step of processing. This asynchronous task processing means that the modules can process tasks as soon they become available, which minimizes idle time.

If you are using an external Microsoft SQL database to store your Intelligent Capture configuration settings and other system data, you can license and configure multiple servers into a ScaleServer group, which acts as a single information capture system. Production modules that are ScaleServer-compatible can connect to multiple servers in a group and receive tasks from all of them. Modules that are not ScaleServer-compatible can connect to one server at a time within a group.

You can also develop custom applications that use the Intelligent Capture REST Service, a Web application, to send documents and data to the Intelligent Capture Server and the Intelligent Capture Module Server, a Windows service. Intelligent Capture Web Client is a browser-based application that uses the Intelligent Capture REST Service.

2.2 Intelligent Capture Real Time Services Architecture

Intelligent Capture Real Time Services is a product offering based on Intelligent Capture REST Services, which are a set of RESTful web service interfaces that custom client applications can use to call the services of the Intelligent Capture Server or the Module Server. An example of an Intelligent Capture REST Services client is Intelligent Capture Web Client.

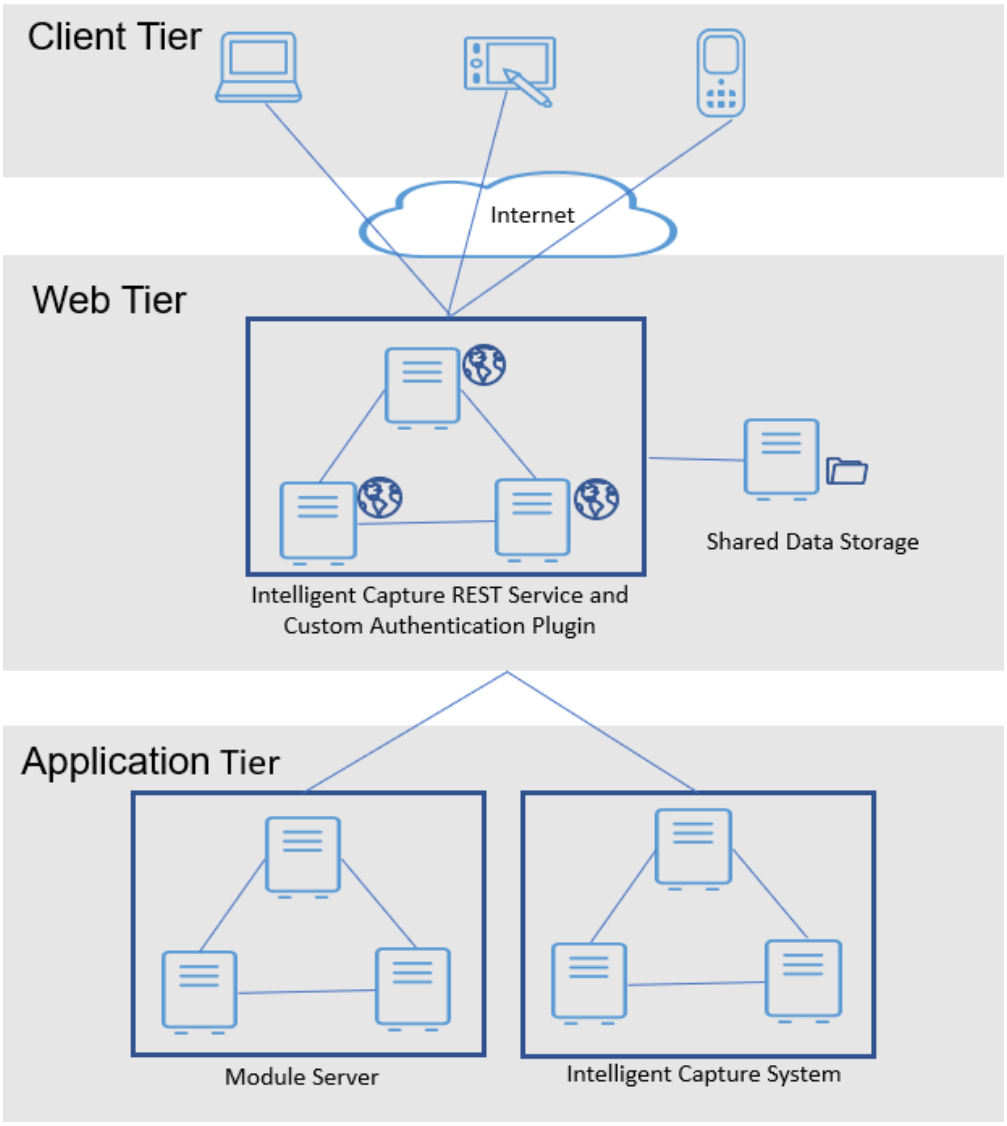
You use the Intelligent Capture REST Services in your application to perform a batch request in a CaptureFlow or an Ad Hoc Service request for Module Server services as follows:

- In a batch request, your application sends documents and data to the Intelligent Capture REST Service Web application, which creates an Intelligent Capture batch, adds the documents and data to the batch, and then sends the batch to the Intelligent Capture Server, which executes the specified CaptureFlow. You can also write a custom Intelligent Capture REST Service Web application authentication plug-in that authenticates and maps the Intelligent Capture REST Service Web application's callers to the appropriate Intelligent Capture user roles.
- In an Ad Hoc Service request, your application makes a request to the Intelligent Capture REST Service Web application for Module Server services, such as classifying and extracting pages or reading barcodes.

The Intelligent Capture REST Services architecture is shown in the following diagram.

- The Intelligent Capture REST Services are deployed to IIS.
- The Module Server is a Windows service that provides classification and extraction, full-page OCR, image conversion, and image processing features.

  A Module Server Windows service manages a set of service modules and can be scaled up to meet demand.


Client Tier
Internet
Web Tier
Shared Data Storage
Intelligent Capture REST Service and
Custom Authentication Plugin
Application Tier
Module Server
Intelligent Capture System

2.3 Storing Intelligent Capture Configuration Settings and Other Data

During the installation process for the Intelligent Capture Server, you can choose to install the external SQL Server-hosted database to store Intelligent Capture configuration settings and other data. If you do not install the external database, then settings are stored in a file-based internal database on the server machine.

2.3.1 External Database

You can choose to store configuration settings and processing information generated by production modules in an external SQL Server database. A central location for the storage of certain processed data, security and configuration settings, and logging information enables the administration of multiple servers, whether or not they are configured in a ScaleServer group.

The database stores the following information:

- Configuration settings.

  Administrators can modify settings without interrupting or impacting the processing of data.
- License codes.
- Logging rules that are used to capture errors, audit data, and other values for use in various displays and reports.
- Data on work-in-progress.

  Administrators can view metadata (the lists of batches and their status) without requiring the server to open every batch. This improves performance when viewing metadata.
- Batch settings.
- Web Services subsystem configuration.

For more information about the database, see *OpenText Intelligent Capture - Administration Guide (ECPCORE-AON)*.

2.3.2 File-based Internal Database

If you do not install an external SQL Server-hosted database, then configuration settings and processing information generated by production modules are stored in a file-based internal database on the Intelligent Capture Server. In this scenario, the internal database is embedded within the Intelligent Capture Server and cannot be shared with other Intelligent Capture Servers.

The following features are not supported for the file-based internal (embedded) database:

- Microsoft Failover Clustering support
- ScaleServer support
- Audit Logging and Reporting
- Web Services
- Upgrading from an external database to the file-based internal database

2.4 Design Tool

Intelligent Capture provides a centralized development tool called Intelligent Capture Designer for creating, configuring, deploying, and testing the capture system end-to-end. This tool serves as a single point of setup for process design tasks and enables access to capture process design tools:

- Reusable configuration profiles and document types, which you can apply across capture processes and assign dynamically per task.
- Configurable third-party image processing filters for image quality enhancement and preparation for the extraction step.
- Export images and data. Standard Export can transform batch data to the following formats and repositories: *CSV*, *XML*, free text, data file, email (*HTML*/ Text), CMIS-compliant repository (Content Management Interoperability Standard), and OpenText Content Server.
- CaptureFlows that specify how the batches are created and how the tasks are performed using attended and unattended modules.
- Deployment environment isolating the capture process design and customer projects from environmental factors such as connections and database queries, enabling secure deployment, and minimizing the time spent to perform updates.
- Integrated development environment, which lets you focus on your profile design tasks without switching to other Intelligent Capture tools.

2.5 Design Areas

In addition to providing a single location for setting up and managing everything related to capture process design and configuration, Intelligent Capture Designer allows several users to work on the same capture site and profiles simultaneously. The *OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)* provides additional information.

To facilitate capture system design tasks, Intelligent Capture Designer unites a number of design areas.

Image Processing
Create profiles with filters that enhance image quality, detect image properties such as barcodes or blank pages, and make page corrections such as deskewing and rotating. You can also add and edit annotations on images.

Image Conversion
Create profiles that specify image properties including file format, color format, and compression; convert non-image files to images and images to non-images (for example, TIFF to *PDF*); merge and split documents; and merge annotations added to TIFF images by other modules into the output image.

Standard OCR
Create profiles to extract data from electronic documents and images, convert input files to PDF or Text format, and produce OCR data cache as a result of processing.

Recognition
Create recognition projects that identify the templates, base images, and rules for classifying documents. If you created Dispatcher project (DPP) files using Dispatcher for InputAccel 6.0 SP3, 6.5, or 6.5 SP1 and you have an Advanced Recognition license, you can import them intoIntelligent Capture. Importing these files lets you use the field placements that are already defined on your existing templates.

Document Types
Create a *document type* for each paper form and associate it with a recognition project. The document type defines the data entry form that the Completion module operators use for indexing and validation. Document type definition includes defining fields and controls, a layout, a set of validation rules, and document and field properties. When you save the new document type, an *indexing family* is generated within the recognition project. The indexing family contains all the index fields for all pages of the document.

Export
Create profiles that specify how data should be exported for your capture processes. Export profiles let you export to standard export formats and repositories such as *CSV*, *XML*, free text, data file, email (*HTML*/Text), CMIS-compliant repository (Content Management Interoperability Standard), and OpenText Content Server.

CaptureFlow Designer
Create and design new Intelligent Capture processes. Each process is a detailed set of instructions directing the capture server to route images and data to the appropriate client modules in a specific order.

2.6 Intelligent Capture Modules

The modules that are available as part of the Intelligent Capture platform can be grouped into the following general categories:

Administrator
The Intelligent Capture Administrator is a tool that enables administrators to manage batches, users, processes, licensing, and reports.

Designer
The Intelligent Capture Designer is a centralized capture system development tool that enables developers to design, debug, compile, and deploy a capture system.

Web Client
The Intelligent Capture Web Client adds value to your document capture operations by providing an easy-to-use, Web-based capture application that you can run in your browser at branch offices and other remote locations.

Real Time Services
Intelligent Capture Real Time Services is a product offering based on Intelligent Capture REST Services, which are a set of RESTful web service interfaces that custom client applications can use to call the services of the Intelligent Capture Server or the Module Server. An example of an Intelligent Capture REST Services client is Intelligent Capture Web Client.

Intelligent Capture REST Services include the following components.

- Intelligent Capture REST Service: The Intelligent Capture REST Service Web application is a JSON REST web service that provides batch creation and Module Server processing features.
- Module Server: The Module Server is a Windows service that provides classification and extraction, full-page OCR, image conversion, and image processing features.
  - Operator Tools: Modules that operators use in production mode.
  - Input/Output: Modules that can create batches and save data to standard formats.
  - Utilities: Simple modules that perform specific routine tasks.
  - Image Handling: Modules that can manipulate images and their properties.
  - Recognition: Modules that perform optical character recognition and data extraction.

- Enterprise Export: Export modules designed to store data directly to specific third-party back-end *ECM* systems or databases.
- Web Services: Web Services components that are used with the web services input and output functionality.
- Advanced Recognition: Advanced recognition modules and tools.

Note: The use of some modules might require the purchase of a specific license.

Overview of Intelligent Capture Client Modules and Utilities provides a short description of client modules and utilities. For a detailed description of a particular module, see *OpenText Intelligent Capture - Module Reference (ECPCORE-CMD)*.

2.6.1 Production Modules

Production modules usually run on client machines. Multiple production modules can run on a single machine, or each can run on a different machine. The optimum configuration depends on many factors, including the amount of data to process and the specific modules involved. The *OpenText Intelligent Capture - Installation Guide (ECPCORE-IGD)* contains information to help you determine where to install production modules.

Most production modules are *unattended* modules, and you can set them up to automatically receive and process tasks from the server. Typically they run as Windows services.

A few modules require an operator to perform manual tasks to complete the module processing step. These modules include those in the Operator Tools category as well as the Identification module, which requires the purchase of an additional Advanced Recognition license. Furthermore, the Completion and Identification modules are part of the Intelligent Capture Desktop family. Overview of Intelligent Capture Client Modules and Utilities provides a short description of these modules.

2.7 User Roles

Intelligent Capture is designed around the following main user roles:

- **Designer**: Creates CaptureFlows and capture profiles that define how information moves through the system. Each CaptureFlow serves as a model for a capture process. Typically there are few designers (perhaps only one).
- **Administrator**: Manages day-to-day operations. Administrative tasks include managing servers, assigning user roles, reviewing system logs if necessary, and ensuring that the system is functioning correctly. Typically there are few administrators (perhaps only one).
- **Operator**: Performs one or more manual tasks using production modules. Typically there are a number of different operator tasks, and many operators who perform each type of task.

The system is designed so that responsibilities can be easily divided into these user roles, although it is possible for a single person to have more than one role. More specific operator roles and user rights can be assigned using Intelligent Capture Administrator.

For the purposes of accessing the documentation, no distinction is made between designers and administrators. Users in both roles have ready access to the full set of product documentation. Operators are provided with an easy way to access only the documentation that supports operator tasks. However, this restriction is implemented as a convenience rather than a security measure. For various reasons, additional documentation files might be installed on a client system that is normally used only by operators, and an operator could potentially view this information.

Chapter 3

Processes and Batches

This section describes how processes and batches move data through the Intelligent Capture system.

3.1 Processes

A process is a detailed set of instructions directing the server to route images and data to the appropriate client modules in a specific order.

Creating a process in CaptureFlow Designer includes the following high-level steps:

1. Creating reusable configuration profiles and document types in Intelligent Capture Designer for modules that require these components to execute tasks. Profiles specify configuration settings for processing images while document types define the data entry forms that the Completion module operators use for indexing and validation. After uploading these service components to the capture server, you can use them across multiple workflows.
2. Creating a CaptureFlow, which consists of elements that define how batches are created and processed using attended and unattended modules.
3. Optionally compiling a process and resolving design issues.
4. Installing a process to the test capture server, which saves the last changes to the process file, compiles the CaptureFlow, and uploads an instance of the compiled process with the specified name to the capture server.

   Note: A process can have one or more versions which lets a customer assign batches to a specific version of this process and manage batch execution appropriately if the process needs to be changed later. When you click to install a process, the folder for the initial process version is created on the server. Thereafter, the new version folder is created every time the process is changed and the changes are uploaded to the server. Each process version folder contains *XPP*, *IAP*, and *DLL* process files.

   You can install multiple instances of the same process under different process names.
5. Deploying to the test server all service components that are required by the steps of the installed process in production. The service components may include profiles, document types, recognition projects, and code for the .NET module.
6. Setting up steps of the installed process. Setting up a step implies running the associated module in setup mode and setting its functional parameters.
7. Testing and debugging the workflow prior to production use.

8. If testing was successful, the next step is deploying (uploading) the desired service components to the production capture server. This step installs the current version of the process to the server and synchronizes the local and server versions of the *XPP* file.

   You can view the current status (for example, **Local changed** or **Unchanged**) of each service component (profiles, document types, queries, styles, CaptureFlows, and other service components) you previously created or modified using Intelligent Capture Designer.

3.1.1 Department Routing

Department routing is a feature that enables tasks to be routed to specific module instances. Within a process, departments can be defined as static per-step values or can be defined dynamically by setting one of two reserved IA values: `IATaskRouting` (to perform task-level routing) or `IADepartments` (to perform step-level routing). Departments can also be assigned dynamically at runtime by the module or the module operator. Module operators use a command line argument that specifies one or more departments. Thereafter, that module instance only receives tasks that belong to the departments specified in its startup command. For example, only operators starting Completion with the "AdminReview" department receive tasks whose current department is "AdminReview". Other operators cannot process those tasks. Additionally, Completion operators can choose departments from within the module while it is running.

When using dynamic routing, the IA value must be set before each step in the process that uses departments. For example, if your process consists of:

```
ScanPlus > Completion > Standard Export
```

and you set `IATaskRouting` to "AdminReview" for Completion, it does not remain set for Standard Export. This module processes all tasks if they are started normally (with no *`-department`* specification). If you want other modules in your process to pay attention to the department value, set it for each module step as needed.

Department routing can route tasks based on conditions that the workflow detects. For example, a condition can be the language, operator security clearance, or service level agreements. Use departments when pages are processed in multiple languages that use multiple code pages and some modules run on a machine configured for a specific code page. In this case, departments can route tasks to modules running on separate machines, each configured with the appropriate system code page for the language it processes.

Identify conditions by:

- Indexing entries made by the ScanPlus operator.
- Characteristics such as document classification.
- Bar code recognition.
- *OCR* results.

- Level changes in the document structure.

Conditions can also be determined by a manual workflow process, such as controlling the sequence of documents that correspond to each department definition.

For security purposes, if needed, *ACL*s can be applied to departments to specify which users can access each department. Users who start a module using a department name to which they do not have access do not receive tasks for that department. Department ACLs enable you to control access to sensitive information by routing tasks to departments that have a restricted set of users. Department ACLs are defined in the **Departments** pane under **Systems** in the navigation panel of Intelligent Capture Administrator.

Intelligent Capture provides two levels of department routing: task-level routing and step-level routing.

- *Task-level routing* is defined at the task level. Each task can be routed based solely on its department name. Use task-level routing when you want to control which operators receive which tasks. In task-level routing, the application routes each task by setting the task-level IA value `IATaskRouting` to a department name. Tasks associated with a specific department are sent only to modules that specified a matching department name when they were started in production mode.

  For example, an operator who is fluent in French starts the Completion module using the "French" department. The operator receives tasks that the *IPP* identifies as French. Tasks in other languages are routed to operators who start modules using departments such as "Spanish", "Chinese", or "Italian".

  Note: If you were to use step-level routing for this type of routing, you would need to define a separate Completion module step for each language, design logic to route tasks to the appropriate step based on the language value, and set up each Completion module step independently.

- *Step-level routing* is defined at the step level. In step-level routing, batches are routed according to the setting of a static Department Name or by dynamically setting the step-level IA value `IADepartments`. Step-level routing with static department settings is ideal for load balancing, where control over urgency of processing is needed.

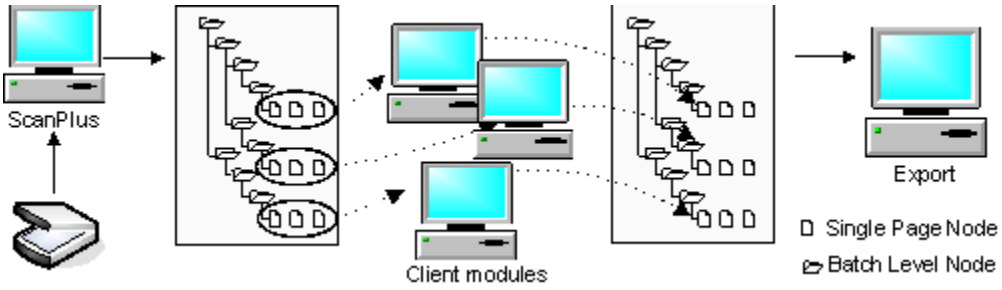
  For example, define one static department, "Urgent". When workloads increase and there is a deadline, use this department to route work to additional operators. To spread the workload evenly, the additional operators must start their Completion modules using the "Urgent" department.

Defining departments in a process depends on whether you are using Intelligent Capture Designer or Process Developer. For more information on how to define departments, see *OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)* and the *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*. Choose the guide that is appropriate to your development needs.

3.2 Batches

The Intelligent Capture platform captures information for processing and digital storage in collections called *batches*. A batch is created by selecting a process that contains appropriate instructions for the data to be processed, and then importing the data. The created batch is always based on the latest version of the selected process.

Batches can be created using data from various sources. A typical batch starts as a stack of paper that gets scanned into the system and converted to image files. Each original page becomes a *node* in the batch. Pages can be grouped and organized into a tree structure of up to eight levels, where the pages themselves are at level 0 (the bottom), and the batch as a whole is at level 7 (the top).



The batch data moves from module to module as determined by the processing instructions. A module might process all of the batch data at once, but it is more common for the data to be separated into smaller work units, or *tasks*, for processing. In the language of CaptureFlows, this means that the batch is processed at a level lower than 7. In many cases, data is passed at the page level, so that each task involves processing only a single scanned page.

Batches can be created using administration tools, but they are usually created directly by an import module. A ScanPlus operator is often responsible for creating batches.

3.2.1 How Batches Are Processed

Batches are created and stored on the capture server. The server controls batch processing, forms the tasks and routes them to available modules based on the instructions contained in the batch.

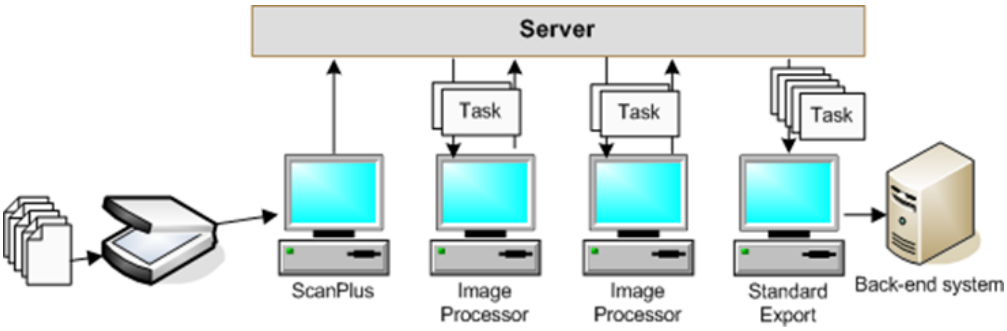
All batches on the server are queued and processed according to their *priority* (0 being the lowest and 99 being the highest). The batch priority is defined by the process settings when the batch is created. If not specified, the batch priority is set to 50 by default. A 'zero' batch priority excludes the batch from processing. Batches that have the same priority are processed according to creation date and time.

The server monitors all the client machines and sends them tasks from any open batch. If multiple machines are running the same module, the server sends the tasks

to the *first available module*. The batch node used by the task is locked when it is being processed and is unavailable to other modules.

In a ScaleServer group, each batch exists on only one server in the group. Each server does its own task scheduling without coordinating with other servers.

When a module completes a task, it returns the task to the server and starts processing the next task from the task queue located on the module's client machine. When the server receives the finished task, it includes the batch node of that task in a new task to be sent to the next module in the CaptureFlow. It also sends a new task to the module that finished the task. If no production modules are available to process the task, then the server queues the task until a module becomes available. This exchange is made possible by*trigger IA values* which signal the server to send a task to a module for processing. The server and the modules work on a "push" basis. The push of data takes place when all the trigger values are set, which typically is when the name of a stage file stored in an *input* or *output IA value* is passed to the next module.



Each item at each level of the batch tree is called a *node*. As the server pushes tasks through the system, the nodes in the tree hierarchy are updated with information stored in IA values that store metadata and enable the system to track processing status.

3.2.2 Batch Files

Each batch has an ID, which is a 32-bit integer that is unique within a ScaleServer group.

The data for each batch is stored on the server in its own hierarchy of folders. The folder hierarchy is based on the digits of the batch ID divided into three groups of digits, split 4-3-3. For example, a batch whose ID is 0123456789 is named `0123456789.iab` and is stored in the folder `Batches\0123\456\789`.

The batch folder contains:

- **Batch file**: Contains the batch tree structure and all IA values. As batches are processed, IA values are updated with the value data generated by each module. The file extension for a batch file is IAB.
- **Recovery file**: An empty text file named with a global unique ID (*GUID*).

- **Stage files**: Data files.

  For each page scanned or imported, a module sends one or more data files to the server. A page is defined as a single-sided image. When a physical sheet of paper is scanned in duplex mode, it results in two pages (one for each side).

  Typically, one stage file is created for each page scanned or imported. However, some modules create multiple files per page. The type of file in which page data is stored varies depending on the module. Supported file formats are described in *OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)*.

  Each stage file is associated with a node and is named with the unique node ID, along with a filename extension corresponding to the stage number during which the file was created. The stage number is sequential according to the order of steps in the CaptureFlow. For example, if ScanPlus is the first module, image files that ScanPlus sends to the server are stored with the file extension 1. Stage files from the next module would be stored with the file extension 2. The numbers are written in hexadecimal format. As an example, if the node ID is 23e, the names of the stage files are 23e.1, 23e.2, and so forth. The server can store a maximum of 255 stage files per node.

  Note: If the input device outputs multiple streams (for example, a multistream scanner that outputs a binary and color image for each page scanned), then each stream is treated as a stage. This means that two sequential file extensions such as 1 and 2 could belong to the same CaptureFlow step. Files created by the next module would then be saved with the file extension 3.

The following table shows a sample record structure for a node with ID 23e in a simple linear process consisting of three modules. The stage files are created when the server receives the stage file name stored in the `OutputImage` IA value of each module step.

Table 3-1: Stage File Values in a Node Record Structure

Module	IA Value	Value Data
ScanPlus	`OutputImage`	`<ca:9c-23e-1`
Image Processor	`InputImage` `OutputImage`	`<ca:9c-23e-1` `<ca:9c-23e-2`
Completion	`Image`	`<ca:9c-23e-2`

The value data `<ca:9c-23e-1` is interpreted as follows:

– `<`: Designates a stage file.

– `ca`: Identifies the client and server communication session.

– `9c`: The batch ID.

– `23e`: The node ID.

– `1`: The stage number.

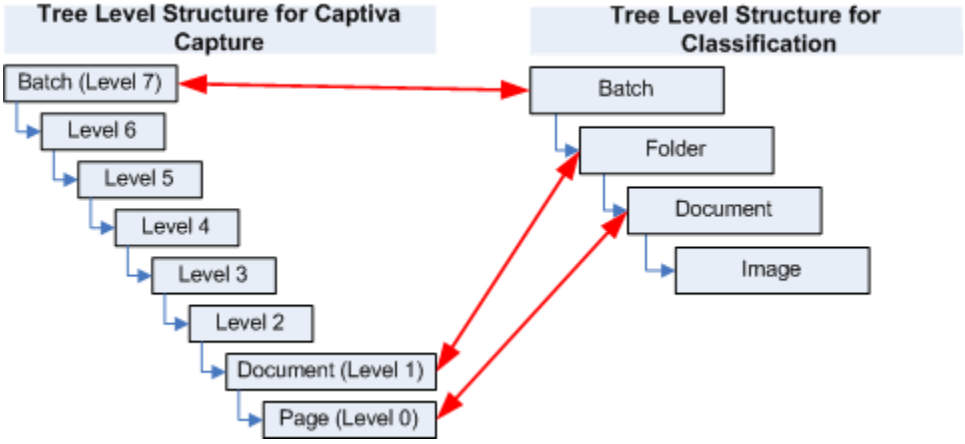
3.2.3 Understanding Processing Levels

This section explains the difference in the processing levels between recognition modules and other Intelligent Capture modules. An example of the relationships between different processing levels is available at the end of this section.

Unlike other Intelligent Capture modules, Classification historically works with batch structures that have only four processing levels as follows: *batch*, *folder*, *document*, and *image*.

When the recognition is done, the data arranged in a recognition structure must be saved on the Intelligent Capture Server. This includes mapping the recognition levels on the standard Intelligent Capture batch processing levels. In case of Classification, this mapping is transparent to the user and does not require any special configuring of these modules.

The recognition levels are mapped to the standard Intelligent Capture levels as shown on the diagram to follow.



The following process is depicted in the above graphic:

- The *image* level in the recognition structure stores templates (sample documents), each presenting a certain kind of a document, such as an invoice, a claim, and others. The Classification module uses templates to classify scanned images and to extract data from certain image areas. The Classification module uses templates of the *recognition project* that is pointed in the module setup settings.
- The *document* level in the recognition structure is used to store one scanned image. The recognition document level is processed as the page level (level 0) in Intelligent Capture.

  If a document was scanned on both sides in recognition, there are two '0 level' pages in Intelligent Capture.

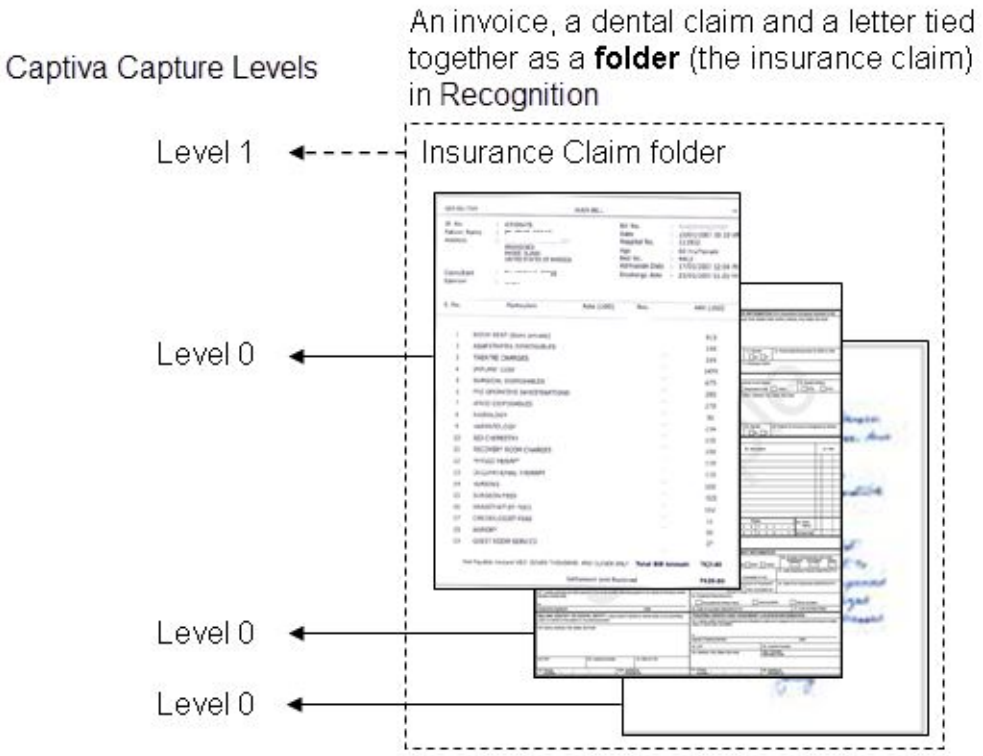
  To process recognition documents that are composed of several images (multi-page documents), templates must be set up to process all images in recognition at page level (level 0) in Intelligent Capture.

- The *folder* level is used in recognition to store several recognition documents as one transactional unit. For instance, a folder may include all documents with the same value in a certain index field, or a new folder may be started when a certain page (a separator) is scanned.

  A recognition folder equals the Intelligent Capture document level (level 1).

Example of Relationships between Recognition Levels and Standard Intelligent Capture Levels

For example, consider an insurance claim that includes a dental claim form, the dental clinic invoice, and a letter from the patient. All these three documents compose a transactional unit: the insurance claim. In recognition, set up a method how to assemble all three documents associated together in a single folder. Create two templates: one for the dental claim form (probably a graphic template) and one for the invoice (probably a free form template). Associate each template with an index family to carry out data extraction from image. Set up handwritten classification without data extraction for the patient letter. In Intelligent Capture, the documents are associated at the document level (level 1) and each document is processed for classification and data extraction at the page level (Level 0).



3.2.4 IA Values

An *IA value* is a variable that is used to store data in a process. It carries information from module to module. IA values can also control when tasks are processed.

IA values can be manipulated by processes in many ways. For example, based on the contents of an index field, a process can determine that a page routes to a Chinese *OCR* step rather than a French OCR step.

IA values that a module defines and exposes to other modules are referred to as *production IA values*. Modules expose their production values by declaring them in a Module Definition File (*MDF*). Production values typically include task-related input and output file values, module data values, and statistical values.

Categories of IA values include:

- **Input and output files**: IA values that hold pointers to the files the module creates, receives, or sends within a task. The files are stored on the server along with other batch files. As with data values, input and output file values are used to "connect" module steps together. For example, to create a simple process with ScanPlus and Image Processor steps, the `InputImage` value of the Image Processor step is set equal to the `OutputImage` value of the ScanPlus step. The difference with images is that, for images, the string that identifies the image is duplicated but not the image itself.
- **Trigger values**: A subset of input file values and client processing values that are used to kick off processing when specific conditions are met. In almost all cases, the `InputImage` or `InputFile` value is a trigger. For example:

  1. An export module triggers at level 7 and uses only the `InputImage` value as a trigger.
  2. The upstream modules finish processing tasks and the `InputImage` values are set to non-zero value data.
  3. The export module starts batch processing because the trigger condition has been met.

  In addition to `InputImage`, modules use other trigger values that control task processing. Review the IA value topics in the *Reference* section of each module's guide for a complete list of IA values.
- **Setup values**: Module step configuration and setup values, such as scanner settings, image settings, OCR language settings, index field definitions, and many others. The settings can potentially change for every task the module processes. For example, you can have ten machines that are running Completion and that are all configured to accept tasks from any batches being processed. Since the tasks from different batches can have different index fields, the settings needed for each task received are potentially different. The module displays the correct set of index fields for each task it receives because the setup values are sent with the task.
- **Client processing results and statistics**: IA values that hold all of the metadata that results from processing tasks in each module. For example, most modules

have IA values that hold date and time an image was scanned, operator name, and elapsed time to process a task. Specific modules can also have values for index field contents, OCR results, and error information.

- **Batch values**: IA values that are related to the nodes in a batch. These values can be created dynamically during processing. For example, a process can include code that finds the number of pages in the current batch and stores it in a batch value named <*MyPageCounter*>. If a value by that name does not already exist, then it is automatically created the first time it is set.
- **Non-MDF values**: Special IA values that hold the following items: Batch name, ID, description, priority, and process name. These values also appear as the titles and entries when creating a batch.
- **System values**: IA values that are related to user preferences, hardware configurations, machine names, and security. In most cases, system values are global in scope and do not apply to tasks contained within a batch. System values are referenced by strings and include: `$user`, `$module`, `$screen`, `$machine`, and `$server`. For example, when a module stores a file that is not associated with a particular batch or process, it uses the "$module" key to store and retrieve the file from the server. An example of this type of file is an OCR spell-checking dictionary.

Production IA Values

Production IA values are values that a module exposes to other modules. Modules expose their production IA values by declaring them in a Module Definition File (*MDF*). Production IA values typically include task-related input and output file values, module data values, and statistical values, but typically do not include global values. An MDF is a text file that contains a declaration for each defined IA value. When a process is defined, the MDFs of the modules used in that process are included. Consequently, all of the IA values in the MDFs are available to the process code. The process code can use the IA values as needed. Each module can see the production IA values that are defined in the MDFs of all the other modules.

Production IA values can be of the following data types: String, Long, Double, Date, Boolean, Object, or File. IA values are declared in MDF as Input or Output values (or both at once) to indicate if the values input to a module or the module outputs values. IA values can also be declared as trigger values. Any trigger declared in an MDF is only used as a trigger if it is referenced in the *IPP* file. All such referenced trigger values must be initialized with data (non-zero) before the module can process the task with which they are associated.

Production IA values are associated with a particular node level. A given IA value can be declared in MDF at only one level. However, because some modules can be configured to trigger at different levels, IA values can also be declared at the trigger level (level T), enabling the IA values to apply to whatever level the process specifies as the module step's trigger level.

Different classes of modules declare different types of production IA values in their MDFs. For example:

- **Task creation modules**: The first module in a process that creates batches from a specified process and starts the document capture job. Typically, task creation modules can also open existing batches when necessary. Task creation modules include ScanPlus, Web Services Import, and Standard Import. These modules do not use input IA values because they do not receive tasks from other modules. However, task creation modules use output IA values for storing data captured during batch processing and statistical data about the batch processing.
- **Task processing modules**: Accept tasks from other modules, perform an operation on the data in the tasks, and then send the tasks to other modules. Task processing modules wait for any task from any batch or open a specific batch to process its tasks. Task processing modules include RescanPlus, Completion, Image Processor, and many others. These modules use input IA values to obtain data from other modules and output IA values to make data available to other modules after the module completes its processing.
- **Export modules**: Obtain the results of document capture jobs out of the Intelligent Capture system and into a longer-term storage solution. Depending on the export module, the destination for exported data can a file system, a batch, or a third-party repository. Modules designed to export directly into a repository can map IA values to the object model of the target system. Images and data files, statistical data, index values, and bar code values can be mapped to the appropriate objects.

Chapter 4

System Administration

Use Intelligent Capture Administrator to monitor, configure, and control an Intelligent Capture system. An administrator can view and configure aspects of the system relating to:

- CaptureFlow definitions
- Batch data (in real time as it is processed)
- User departments, roles, and permissions
- Servers and ScaleServer groups
- Web Services configuration
- Licensing

  Note: Intelligent Capture REST Service client (including Intelligent Capture Web Client) and Module Server licensing is managed through the Intelligent Capture REST Services Licensing tool.
- Logging and reports

For more information on administering the system, see *OpenText Intelligent Capture - Administration Guide (ECPCORE-AON)*.

Chapter 5

Security, Performance, and Scalability

Security, performance, and scalability are key concerns in most enterprise environments. This section summarizes the features of Intelligent Capture that address these concerns.

5.1 Security

Intelligent Capture security is managed through Intelligent Capture Administrator roles and Access Control Lists (ACLs). In general terms, role permissions are for actions and ACLs are for things. Users or groups can use both, but generally speaking, roles are at the top level of securing the system and ACLs are for finer-grain control. Roles contain two important traits: permissions, and users or groups. A role will have a defined set of permissions that are appropriate for members of that role. Each member (user or group) of that role will inherit the assigned permissions.

ACLs define access for users or groups to modules, batches, departments, or processes. They enable administrators control access to these items, separate from role definitions.

Intelligent Capture users and groups are made available to Intelligent Capture Administrator as Windows-defined users or groups. Some of the security in Intelligent Capture is provided by Windows. For example, a user may have permission to run modules and processes in Intelligent Capture, but if these operations require writing to a folder, the user must also have the appropriate Windows rights.

For information about configuring security settings, see *OpenText Intelligent Capture - Administration Guide (ECPCORE-AON)*.

5.2 Performance

An administrator or designer can configure various settings for enhanced performance. For example, image handling modules can use different color compression settings to enable the best balance among performance, image quality, and disk usage.

Intelligent Capture provides performance-monitoring features such as performance counters and statistics reports. Performance counter objects are available only on the machine where the Intelligent Capture Server is installed. Note that some performance monitoring features might require additional licensing.

For information about configuring performance settings and using the performance tools, see *OpenText Intelligent Capture - Administration Guide (ECPCORE-AON)*.

5.3 Scalability

Intelligent Capture is a global, scalable solution that can use multiple servers to manage resources.

5.3.1 ScaleServer Groups

A ScaleServer group is a group of Intelligent Capture Servers that share processing responsibilities. ScaleServer technology provides many benefits including increased availability, higher productivity, improved workload balancing, and centralized control. In a ScaleServer group, up to 8 servers work together as a single information capture system, distributing the processing workload. When each batch is created, it is assigned to one of the servers in the group. Each server manages its own work. Once a batch is assigned to a server, that server manages the batch through its entire processing cycle.

The multiple servers in a ScaleServer group appear as a single server to a production module. If a server becomes unavailable, modules can continue to process tasks from batches on the other servers. Servers share connection information, so a module consumes just one connection license regardless of how many servers are in the group.

Licensing also controls the number of pages that can be processed in a specified time period. To increase productivity and throughput,Intelligent Capture allows individual servers to share pages with other servers in a ScaleServer environment instead of becoming unavailable when their page count allotment runs out. The servers perform page sharing without impacting the client module.

For information on licensing, configuring, and using ScaleServer technology, see *OpenText Intelligent Capture - Administration Guide (ECPCORE-AON)*.

5.3.2 Language and Locale Support

Intelligent Capture handles data in multiple languages and can be used with multiple locale settings.

- Production module user interfaces are available in multiple languages. Modules can run in most system locales, even if there is not a *UI* translation available in that language.
- Dynamic IA Values can store Unicode text data.
- NuanceOCR and Extraction can extract text from documents in many languages.
- NuanceOCR can output Unicode text files. Unicode files are ideal for the global exchange of information because they can contain characters from many languages.

The *OpenText Intelligent Capture - Administration Guide (ECPCORE-AON)* provides additional information about multiple language support.

Chapter 6

Customization Options

You can change the behavior of Intelligent Capture with custom code.

Profile Scripting

Use for document types and page-level image enhancements. This type of script is meant to be used, and reused, across processes. Therefore, it is completely independent of the process, batch structure, tasks, and so on. There is no direct access to IA values or to the batch or process. Profile scripts should never access the task scripting *API*s or events. The *Profile Scripting* section of the *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)* provides information.

Task Scripting

Use for task and batch node manipulation. This type of script has knowledge of IA values and the batch. If it gets a document data block, then a task script can use profile script APIs to manipulate the object. It cannot use the profile scripting events or UI-related APIs. The *Task Scripting* section of the *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)* provides information.

Client-Side Scripting

Use for creating client scripts to automate tasks in capture processes. A client-side script is a program that runs as part of a module step within a CaptureFlow. Several modules support client-side scripting. To use client scripts, you create script actions and then associate them with specific events that are defined in each module. The occurrence of the event triggers execution of the script action. Client-side scripts are handled directly by the modules that support them and do not require an extra step in the CaptureFlow. The *Client-Side Scripting* section of the *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)* provides information.

Recognition Scripting

Use for customizing Classification and Identification to suit specific project requirements. The *VBA* used in these modules adds a VBA-compatible VBA Script Editor and debugger to your application, enabling the language to be extended with user-defined statements, thus enabling end-users to control their applications. It provides a complete integrated development technology, ideal for rapid customization and integration purposes. The *Recognition Scripting* section of the *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)* provides information.

Intelligent Capture REST Services

Intelligent Capture Real Time Services is a product offering based on Intelligent Capture REST Services, which are a set of RESTful web service interfaces that custom client applications can use to call the services of the Intelligent Capture Server or the Module Server. An example of an Intelligent Capture REST Services client is Intelligent Capture Web Client.

Chapter 7

Overview of Intelligent Capture Client Modules and Utilities

This section provides a brief description of Intelligent Capture client modules and utilities.

7.1 Operator Tools

Operators use the following modules in production:

Identification

Enables operators to assemble documents, classify document pages to page templates, verify and edit values in pre-index fields, check and edit images, flag issues, and annotate pages. Permissions for particular operations are determined during module setup. The view and behavior of the user interface is determined during module setup and in global configuration options.

Upon launching the Identification application, operators choose work from the list of batches available for processing. After getting either a single batch or all batches, operators cycle through each task until work has been processed.

Completion

Enables operators to assemble documents, index and validate data, check and edit images, and flag issues. The user interface components that operators see in validation view are determined during module setup and in global configuration options. Document types created in Intelligent Capture Designer determine the appearance and behavior of the data entry form that operators use for indexing and validation.

Upon launching the Completion application, operators choose work from the list of batches available for processing. After getting either a single batch or all batches, operators cycle through each document until all work items have been processed. The types of work items to be addressed for each piece of work are determined by the work level and other Completion setup settings.

ScanPlus

Enables operators to create batches and scan or import pages into them, automatically creating a batch hierarchy based on detected scanning events.

RescanPlus

Enables operators to scan new images to replace those that have been flagged for rescanning. Only pages that need rescanning are reprocessed, not the entire batch. Rescanned pages are positioned in their original place in the batch.

7.2 Input/Output Modules

The following modules can create batches and save data to standard formats:

Standard Import

Import profiles specify the following items for import:

- Image files from directories
- Email and attachments from an email server

The Standard Import module performs the actual import.

Standard Export

Exports content to emails (*HTML*/text), files (*CSV*, *XML*, free text, and data file) and repositories (CMIS-compliant repository (Content Management Interoperability Standard), and OpenText Content Server). A single export step defines the batch data to export, the format for the batch data, and the location where the batch data is written.

ODBC Export

Adds, retrieves, and updates content within supported databases using an *ODBC* connection.

Web Services Input

The WS Input module is an Intelligent Capture client module that functions as a web service provider. A step of the WS Input module can be configured at the beginning or in the middle of a process. When used at the beginning of a process, the WS Input module creates new batches as it receives web service requests from external systems. When used in the middle of a process, the module can insert data and files into an existing batch. The WS Input module provides mapping for simple parameters (single values, structures, and arrays), and it provides client-side scripting on page 37 capabilities to enable processing of more complex parameters.

The WS Input module operates under the control of the Web Services Coordinator and Web Services Hosting components. Before using the WS Input module, the Web Services Subsystem must be configured by using Intelligent Capture Administrator.

WS Output

Web Services module that functions as a web service consumer. A WS Output step is configured at or near the end of a process, enabling the module to export data that has been processed by other modules. By using the WS Output module, customers can extract images, files, and metadata from an Intelligent Capture system to any web-service enabled, third-party system without writing a custom export module.

The WS Output module runs independently and does not rely on the other components in the Web Services Subsystem. Therefore, no configuration is required in Intelligent Capture Administrator.

7.3 Utilities

The following utility modules perform specific tasks:

- **.NET Code**: Runs custom code as an independent step within a process. A .NET Code step can be added to the process like any other module step. The module provides a Microsoft .NET Framework programming interface that can be used to read and write batch data. A developer accesses this interface by creating a .NET assembly (*DLL* file). The .NET Code module's programming environment also provides access to built-in .NET Framework interfaces.
- **Copy**: Automatically copies batches to another capture system, to a local or network directory, or to an *FTP* site.
- **Multi**: Enables processes to manipulate the batch tree by inserting or deleting nodes and/or changing the effective trigger level of a module instance.
- **Timer**: Triggers other modules to start processing tasks from specified batches at a particular time.

7.4 Image Handling

The following modules enhance, manipulate, and add annotation data to images:

- **Image Converter**: Identifies and processes both image and non-image files. Converts non-image documents into image files for processing and export by other Intelligent Capture modules. Splits multi-page image files into single page image files and converts image file format, compression, and color depth to specified values.
- **Image Processor**: Applies image filters to detect data, remove image objects, adjust colors, improve line quality, and correct page properties using Image Processing profiles. In addition to cleaning up scanned images, you can add or edit annotations for images.

7.4.1 Controlling Image Formats

The color compression setting choices configured in setup mode for modules such as ScanPlus, RescanPlus, and Image Processor, affect the image quality, file size, processing speed, and module performance. For example, the *JPEG* color compression format efficiently compresses images into smaller files, but reduces image quality. If the same JPEG compressed image is modified by several modules, then the compounded compression losses might reduce image quality to an unacceptable level.

7.5 Recognition

The following recognition modules perform optical character recognition and image data extraction:

- **Extraction**: Extracts field data into a document type object, which serves as input to the Completion module. Completion uses this object to identify the document type and index fields for the document. It then retrieves the index values from the document type object and pre-populates the data entry form using data from the recognized pages. Operators can then verify the accuracy of the extracted data.
- **NuanceOCR**: Performs full-page optical character recognition of scanned or imported images using engines from Nuance. Exports the image and index data to more than 25 different word processing and text formats.
- **Standard OCR**: Performs data extraction from electronic documents and images by running an appropriate OCR engine processing mode for each type of content. Produces OCR data cache as a result of processing.

7.6 Enterprise Export Modules

The following export modules are designed to store data directly to specific third-party back-end *ECM* systems or databases:

- **Archive Export**: Exports content to a supported content repository using BC-HCS (*HTTP* Content Server) and exports administrative data to SAP R/3 using BC-AL (Archivelink).
- **OpenText ApplicationXtender Export**: Exports documents and data directly to an OpenText Documentum ApplicationXtender system.
- **OpenText Documentum Advanced Export**: Exports documents and data directly to OpenText Documentum Server.
- **FileNet Content Manager Export**: Exports documents and data directly to FileNet Content Manager.
- **FileNet Panagon IS/CS Export**: Exports data directly to a FileNet Panagon Image Services (*IS*) or Content Services (*CS*) system.

  Note: IBM no longer supports FileNet Content Services. As a result, the FileNet Panagon IS/CS Export module no longer supports exporting to the CS system.

- **Global 360 Export**: Exports documents and data directly to a Global 360 Server.
- **Export for IBM Content Manager**: Exports documents and data directly to IBM Content Manager for Multiplatforms.
- **Export for SAP Archive and AP Connect**: Exports documents and data to *SAP* using IBM CommonStore.

- **Microsoft SharePoint Export**: Exports documents and data directly to Microsoft Office SharePoint.
- **Export for OpenText Content Server**: Exports documents and data directly to an OpenText Content Server.

7.7 Web Services

The following Web Services components are used with the WS Input and WS Output modules:

- **WS Coordinator**: Implements web requests management.
- **WS Hosting**: Serves client web requests.

7.8 Advanced Recognition

The following Advanced Recognition modules require the purchase of an additional license:

- **Classification**: Performs image classification.
- **Identification**: Enables operators to perform manual image classification for documents that were not automatically classified by the Classification module.
- **Collector**: Stores automatically processed documents tagged as collectable to create templates learned by Production Auto-Learning Supervisor service.
- **Production Auto-Learning Supervisor**: A service that performs automatic template creation and field positioning based on collected documents.

Glossary

ACL

Access Control List

API

Application Programming Interface

BPM

Business Process Management

CS

Content Server

CSV

Comma Separated Variable

DLL

Dynamic Link Library

ECM

Enterprise Content Management

ERP

Enterprise Resource Planning

FL

Florida

FTP

File Transfer Protocol

GUID

Global Unique Identifier

HTML

HyperText Markup Language

HTTP

Hypertext Transfer Protocol

HTTPS

Hypertext Transfer Protocol Secure

IAP

InputAccel process file extension

IPP

Integrated ProcessFlow Project

JPEG

Joint Photographic Experts Group

MDF

Module Definition File

OCR

Optical Character Recognition

ODBC

Open Database Connectivity

OH

Ohio

PDF

Portable Document Format

SAP

System Analysis and Program Development

SOAP

Service-Oriented Access Protocol

TCP/IP

Transmission Control Protocol/Internet Protocol

UI

User Interface

URL

Uniform Resource Locator

US

United States

VBA

Microsoft Visual Basic for Applications

XML

Extensible Markup Language