Kofax eFlow
Design User's Guide

Version: 6.1.0

Date: 2023-08-29

KOFAX

Table of Contents

Preface

This guide explains how to use the Kofax eFlow Design module to create, set up, configure, customize, and edit applications for the eFlow system.

The eFlow unified content platform is a common gateway for business-critical data entering your enterprise. eFlow's various modules import paper or electronic text data, process it using sophisticated OCR engines and image processing tools, and export the data in customized formats to your database.

> ⓘ This module is used as a tool to design eFlow applications, hence it is also called as Design tool.

Product documentation

To access the full Kofax eFlow documentation set online, see the Kofax eFlow Product Documentation page. For a complete set of Kofax eFlow documents, refer to the Kofax eFlow Release Notes.

Chapter 1

Overview of the Design tool

This chapter provides an overview of the Design tool (also called Design module), some basic concepts, and the basic flow of building an application:

- Designer
- Basic concepts
- Overview of building an application
- Defining OCR engines and virtual engines
- Defining the application workflow
- Running the system

Designer

The Design tool includes three types of designers.

Application Designer

The Application Designer is a one-stop shop for designing eFlow applications.

The Application Designer enables you to define the following:

- An entire application, including all details of the forms it contains. This includes the logical structure of the forms (the list of all the data fields they contain and their attributes) and the physical structure (the actual appearance of the paper form).
- Rules and validation processes for the logical form and its fields, including the definition of the logical relations between fields.
- The data completion method, data lookup tables (dictionaries), and data and exception reports generated by the system.

OCR Engines Designer

The OCR Engines Designer enables you to define OCR engines and virtual engines that are used to recognize characters. You can then test and fine tune OCR recognition parameters.

Workflow Designer

The Workflow Designer enables you to define the application workflow, using templates or custom design.

Once the application, OCR and workflow design steps have been completed (which for simple applications may take only a few hours), the automatic form processing application is ready to start production by running the forms through the system's runtime modules.

Basic concepts

This section explains some basic terms and concepts that you should be familiar with when building eFlow applications.

Flows forms and Empty Form Images

A physical form is the original input - a scanned paper document - that eFlow converts to electronic data as output. A physical form may contain many types of data, such as logos, barcodes, name, date, address, inventory tables, census data, check boxes, and many other types of numerical and text data.

eFlow allows you to associate several physical appearances to a single logical form. Each physically distinct appearance of an empty form is called an Empty Form Image, or EFI. The EFI is a template of the structured document, which the system uses to extract the data. The eFlow term EFI is a logical definition and simply means images of the empty forms.

The following diagram illustrates the concepts of the logical (flows and forms) and physical (EFIs) parts of the application. Flow 1 and Flow 3 use two forms, while Flow 2 uses only a single form. Form 2 is used in two flows, both Flow 1 and Flow 2, and Form 1 is associated with more than one EFI.
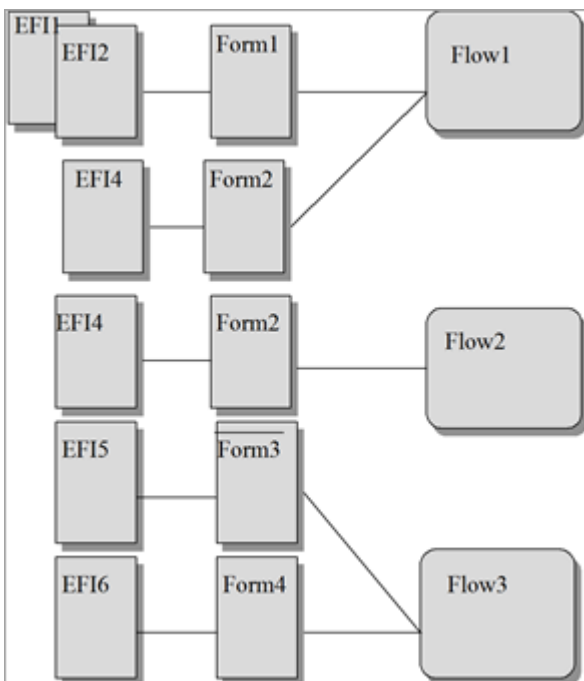
In an application, you define how many flows and forms you want, which EFIs you want to attach to them, and how you want the data processed. You define which OCR engines will be used to interpret the form entries, and what type of image enhancement and form removal parameters to apply to achieve the best results. You can group data items and apply logical and mathematical rules to ensure data correctness and quality, and to reduce the number of exceptions that are returned to the operator for manual data completion.

The form documents to process may be obtained from different sources. Examples of input file sources are:

- Scanned image
- Faxed image
- Text document
- Image file

The digital data output may be sent to almost any data management system that you use. Examples of output file formats and destinations are:

- XML
- CSV
- INI
- SQL
- Oracle DB
- SAP
- CRM

Sample applications are supplied with the system in the folder `…\Sample Applications`, for use as reference applications. However, you should create other applications for your own work and tests.

Forms and collections

Forms can be organized in several ways.

A collection, or batch, is the work unit of eFlow, which is passed from one station to the next. A batch is defined in each station definition. A batch is a set of forms (single or multiple pages) that are separated by a separator (a specially designed page or a blank page). For example, if you want to group sets of 100 forms during input, then inserting a batch separator after such a group will mark it as a batch.

Another way to understand the concept of batches and separators is in terms of how they are scanned in, as shown in the following diagram.
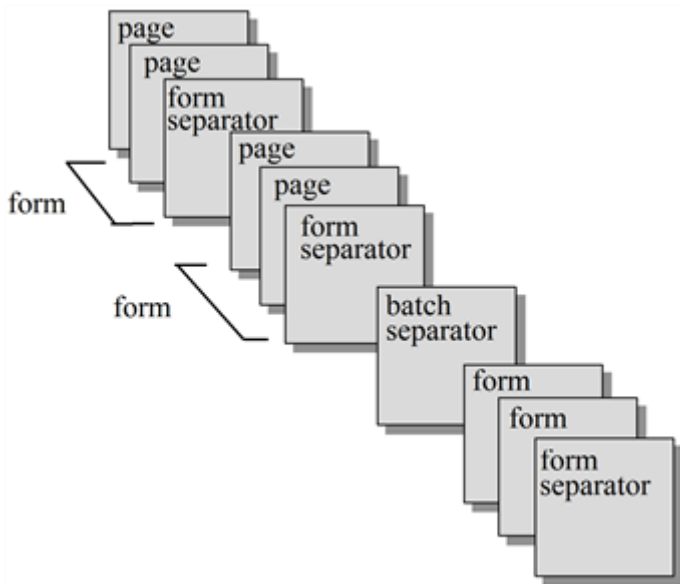
Flows and applications

A flow is a collection of definitions describing processing, enhancement and other parameters that define how batches of forms are identified and processed in the eFlow system.

An application may consist of one or more flows. A flow represents a given path through a workflow. Similar forms may require different processing, some passing through more, fewer, or different stations than others. In this case, you create two or more flows within an application so that different forms are processed according to their differences in format and content.

The eFlow system

Kofax has developed the eFlow unified content platform to deliver the solutions customers require. Whether information enters an organization via predefined forms from clients, internet-based e-forms from customers, incoming supplier invoices, or employees in the field with mobile devices, eFlow digitally captures and manages the enterprise's content from the source through delivery to ERP, CRM, and financial or other systems.

The comprehensive functionality of eFlow means that it is the answer to the steadily increasing amount of paper information and its high costs in terms of time, space, and money. eFlow's exclusive advanced features result in:

- Greater operator productivity
- Improved character recognition and error detection
- Faster and more efficient form processing
- Quick and simple design procedures

eFlow's advanced OCR/ICR technologies achieve unprecedented recognition accuracy. The multi-processing engine architecture enables quick and easy creation of powerful super engines from

basic recognition engines. Coupled with code-free definition of validation rules, human intervention is reduced to a minimum, immediately yielding reduced labor costs and higher data quality.

eFlow products

All Kofax product solutions plug into the eFlow unified content platform and take advantage of the common services that the platform provides.
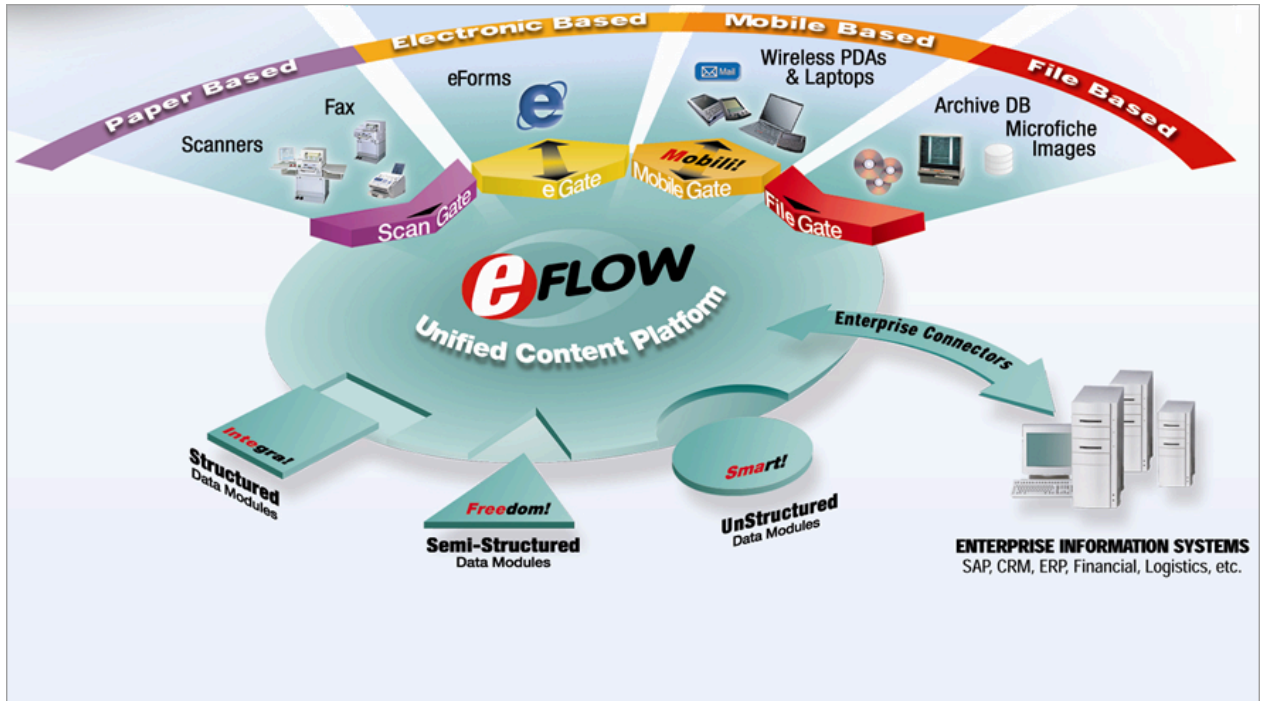
Integra^TM

Integra captures data from all types of structured forms, including paper, fax, image files, the Internet, and mobile devices. Integra automatically validates and delivers the data to the target enterprise application. Integra's exclusive advanced features deliver:

- Greater operator productivity
- Unprecedented character recognition and error detection
- Powerful engines with an exclusive voting algorithm to achieve the most accurate results
- Faster and more efficient form processing
- Quick and simple point-and-click application design
- Minimum human intervention, resulting in reduced labor costs and higher data integrity

Freedom^{TM}

Freedom reads, understands and processes incoming supplier invoices without the need to predefine the structure of all the various invoices. Freedom captures, validates and delivers the data from the invoice to the required applications.

- Handles any type of invoice, including invoices with complicated tables and multi-page invoices
- High-speed content recognition replaces the expensive manual process of keying in data from the invoice
- Recognizes the invoice and its fields according to keywords and key terms situated anywhere on the page, such as payment date, sum, name, and so on
- Unprecedented character recognition and error detection
- Powerful engines with an exclusive voting algorithm to achieve the most accurate results
- Faster and more efficient invoice processing
- Minimum human intervention, resulting in reduced labor costs and higher data integrity

Smart^{TM}

Smart performs fully-automatic classification of documents flowing into an organization.

Smart acts as a mailroom application, deciding which department, business unit or subject each document belongs to, and subsequently, to whom the document should be directed. Smart can classify incoming documents as invoices, change-of-address letters, and purchase orders, or it can classify into more general categories, such as documents belonging to the sales, finance or customer service department. These categories are called classes.

Smart uses a technique called supervised learning to classify documents with little or no prior knowledge about them. The software examines a limited sample of documents from each class, and automatically learns the unique characteristics of that class. This process is called training. From that point onwards, Smart can classify any incoming document into the trained classes, with a high level of accuracy, and with no human intervention.

Runtime modules of eFlow products

The runtime modules are automatic or manual stages in the actual eFlow process as collections of forms move through the system. These modules are realized by "stations", which run automatically or, in the case of manual stations such as Validate or Escalate, operators or supervisors perform specific tasks.

The input images are captured by the scanner interface and transferred to the content recognition module.

Content recognition consists of a general-purpose image processing module and an interface to multiple OCR, ICR, and OMR engines. The results of these engines are processed by a voting mechanism. The final results of the image processing and recognition process are sent, if necessary, to manual data completion. The full results (recognized and completed) are then sent, if necessary, to the Escalate module, and then to export directories. The stored data and images can be retrieved at any time.

The following image shows an example of an eFlow workflow.
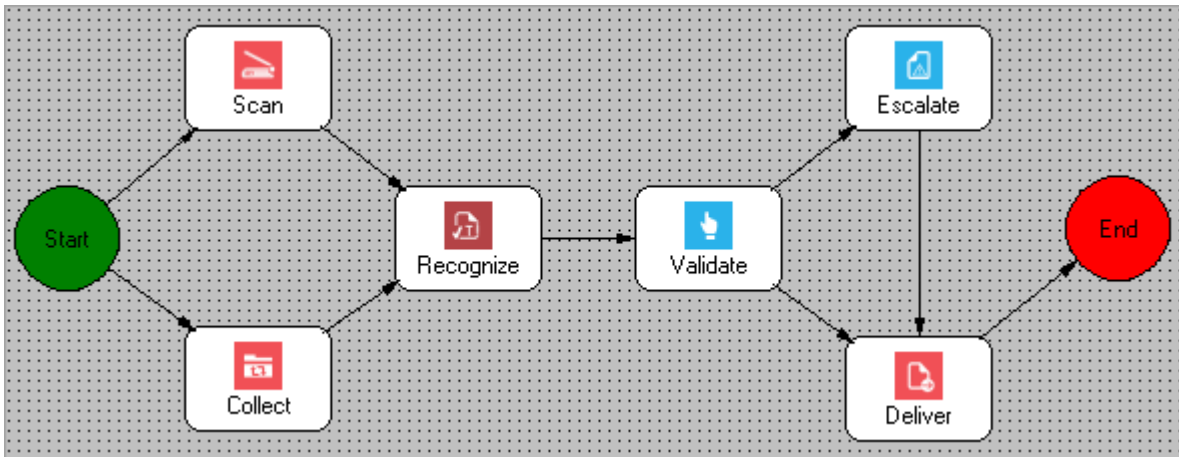
The eFlow Launch module is used to run the modules. Through the Launch module, you operate the data collection, content recognition, content validation, and content delivery modules, which comprise the full data capture process.

eFlow modules, such as Scan, Collect, Recognize, Validate, Escalate, and Deliver, can be realized as one or more instances or stations, depending on the workflow and the size of the project.

The runtime modules are parts of the products. Each product (Integra, Freedom, and Smart) uses the same family of modules; only the modules (executables) vary from product to product.

Category	Module	Description
Data collection	Scan	The Scan module supplies scanned forms to the system. This module supports a wide range of scanners, including those of the Kodak family.
	Collect	The Collect module supplies image files of previously scanned forms to the system.

Category	Module	Description
Content recognition	Recognize	The Recognize module receives data from the data collection modules for form identification. It automatically identifies a form using empty form images supplied during the design stage. This module is one of the most important parts of eFlow. It is responsible for recognizing as much data as possible. The images undergo image enhancement, automatic form recognition, form removal, and optical character recognition.
		Among the many unique features of this module are: separate enhancement algorithms that can be applied to each field and powerful character recognition (OCR and ICR) that is achieved by a sophisticated combination of recognition engines.
		Recognize reads the whole page, using a page recognition engine. It then locates potential keywords and suspected data areas. If keywords are missing (like the state and zip code in an address), the data is recognized according to business and topographical rules (for example, the state is always to the left of the zip code).
Content validation	Validate	The Validate module receives data from the Recognize module. The exact data it receives depends on the design definitions, but typically it is invalid or unrecognized data. Unrecognized data is completed manually, with or without the help of lookup tables. The entire layout of the screen can be customized for one or more stations through the Layout Designer application.
	Escalate	Most forms have rules governing how various fields are filled in. For example, one field may require digits only, while another may require a valid address. Once these rules have been defined, the operator cannot leave the data completion process until they are obeyed. This can be a problem if an entry is illegible or incomprehensible. The Escalate module is designed to deal with these cases. An exception is a mark on the batch that allows the operators to leave invalid or incomplete data and advance to the next form.
	Organize	The Organize module allows operators to delete pages from a multi-page batch, insert new pages into a new multi-page batch (also by scanning), and resort or rescan pages.

Category	Module	Description
	Tile	The Tile module is used for mass validation and decreasing the substitution errors of the recognition engines. It allows for the immediate viewing of recognition engine results grouped in order by alphanumeric character. The Tile module usually receives data produced by the Recognize module, but can be applied anywhere in the system (using the dynamic workflow) according to the specific needs of each project. Grouping the results by fields within flows, for example, by grouping machine print and hand print fields separately, makes it easier to spot false results and send them for correction in the Validate module. Thus, substitutions can be more quickly eliminated.
Content delivery	Deliver	The Deliver module exports data to XML and ASCII formats, and additionally exports both B&W and color images. During runtime, the Deliver module receives the data, converts it to the defined format, and transfers it to the export directory.

eFlow Administrate

The Administrate module manages all the top-level actions of the eFlow systems. Use Administrate to create, copy and delete entire applications, and to manage automatic stations and licenses.

In the Application Designer, the system integrator can define, but not create an application. This action can only be done through the Administrate module, which uses MMC technology (Microsoft Management Console architecture), making eFlow's top level management tools technologically streamlined with the latest, future compatible standards.

eFlow Control

The Control module displays a run-time workflow and its stations graphically and enables users to monitor and control each of the flows of the application. The user can open additional stations where necessary, and prioritize and delete collections, or moving them from one station queue to another.

Scalability

The system is based on a 3-tier architecture for capturing, processing, and storing forms. The network structure facilitates adaptation to changing workloads and functionality by enabling the addition or removal of system components from the network with minimal software modifications (configuration definitions).

Once a decision is made to add stations to the system (for example, more content recognition or content validation stations), the architectural design facilitates an increase in computing and processing power. eFlow and its products are fully scalable.

eFlow's flexible configuration and design allows for easy adaptation to the rapidly changing needs of the computer market. eFlow has an open-ended design that can always be built upon.

Overview of building an application

During system setup, you build a copy of the system at your own site, usually according to the following stages:

1. Prepare the forms that must be processed by the system.
2. Create an application and flow set.
3. Define the forms that will be processed by the system.
4. Define the flows that will be used to process the forms.
5. Define functions to be performed upon events.
6. Define the exceptions that may occur during the data entry process.
7. Define lookup tables for use in the Validate station.
8. Customize the Validate station screen layout.
9. Test the data entry part of the system.
10. Tune the system.
11. Construct interfaces to parts of the system that are not included in eFlow.
12. Test the full system.

These stages are described in more details in the following topics.

Prepare forms for processing

To prepare the forms that will be processed, you need the following:

- Sample empty forms of each type of form to be processed by the system.
- Sample batches of completed forms. These batches should include all cases that will be handled by the system. There should be enough of these samples. Generally, several hundred completed forms are required for each form type.

Create an application

During this stage, you create a new application in the Administrate module, and configure the application in the Design module's Application Designer. An application is the container that includes all the flows, forms, fields, data specifications, lookup tables, rules, and screen layouts; in short, all the logical definitions to be used by the eFlow system.

See Building an application for more information.

Define flows

You can define several flows. A flow describes how the forms are organized in a batch and contains parameters for every stage of the data entry process.

See Design flow for more information.

Define forms and pages

You can define several forms per flow. A form may contain one or more pages. Each page should have its own EFI (Empty Form Image).

See Design form and page for more information.

Define EFIs

Forms are identified by comparing information received through the data collection modules with images of empty forms (EFIs). It is essential to carefully consider what information consistently appears on a form and how forms may be recognized. It is also important to provide appropriate examples for all types of forms that will be processed.

See Design EFI for more information.

Define field groups fields and tables

Form data is extracted from fields and table items. You specify which fields and tables contain data that must be processed. It is important to identify the types of information that are to be expected.

See Design field group and field and Design table for more information.

Define rules validation functions and events

You should define rules and validation functions, such as whether a value is mandatory or appropriate for each field. For example, you may define that there must be a string in a name field. If there is no string, the form will be sent to the Validate station for manual completion. You can define functions and events that can be linked to certain fields in the Validate station.

The flow event mechanism enables you to program the system's reaction to a specified event in the various stations. Sets of possible events are defined for the Scan, Collect, Recognize, Validate, Escalate, and Deliver stations. You can specify functions to be executed when these events occur.

See Define rules and validation functions and Event functions for more information.

Define layouts

During this stage, you customize the entire layout of the Validate screen, for one or more stations, as per the user's requirements. For example, you can build a Validate display layout that will provide tables of critical information in red. You can provide buttons for better functionality. You can add icons to visually remind operators of the expected contents of a field, such as a telephone for a telephone number field.

Define exception categories and exceptions

You then define the types of exceptions (exception categories), and exceptions. The definition of exceptions is also part of form definition. Here, you define all the problematic cases that may occur in the processed forms, and how these cases should be handled.

See Define exceptions for more information.

Define lookup tables

During this stage, you define lookup tables – files that facilitate data entry in the Validate station by providing selection lists of known possible field values. Lookup tables can be associated with one or more forms. Each item in the lookup table is associated with a corresponding field in the form.

See Define lookup tables for more information.

Define OCR engines and virtual engines

This section defines how to select an engine, create virtual engines and test the engines in brief.

Define and select engines

The OCR Engines Designer provides pre-tuned engine templates that can be used for character recognition. It also provides sophisticated parameter control, fine-tuning, and recognition results generation and checking for the OCR engines that you use.

Each engine contains a set of parameters defining how recognition is performed for a field. Each engine can be fine-tuned using the parameter settings.

See Design OCR engine for more information.

Create virtual engines

To improve recognition results, instead of a single OCR engine, you can group engines together to create an OCR virtual engine. A virtual engine contains one or more engines, specifically configured to optimize the recognition capabilities. Each engine has specific strengths in recognition performance. When combining engines in a virtual engine, choose engines that are most suited to the type of data that will be found in the specified region.

Every OCR engine supplies a prediction of each character that it examines, and supplies a confidence level for every character it recognizes. The combined results from the OCR virtual engine are merged using a voting method.

See Design virtual engine for more information.

Test engines

After you have set up new OCR engines and virtual engines, you can use the engine tester to generate statistics regarding engine performance to evaluate the results.

Use the testing results to locate segmentation and recognition problems to improve how engines handle information. Then, using the results that you have generated with the engine tester, change relevant properties, attributes or enhancement methods until results improve.

See Test virtual engines for more information.

Define application workflows

eFlow is fully customizable. You can create, define, and edit a workflow to fit your needs, depending on the project. You create the stations, data routers that determine the flow of data, and actions performed.

See Workflow design for more information.

Run the system

This section describes how to test the data entry system, tune the system and build interfaces.

Test the data entry system

In this stage, you test the data entry part of the system. You test all the stages that operate during the data entry process: input, recognition, data completion, archive, and backup.

Tune the system

There are several aspects of the system that need tuning.

Scanning parameters

You must select appropriate parameters for scanning. A good image should contain data without noisy areas, and with unbroken lines and text. Tuning the scanning parameters correctly is crucial to the performance of the whole system.

Form recognition

The first thing that the Recognize station does is to identify every page of every form in the batch. Sometimes, the results of the identification during the first try are not satisfactory.

To improve the form identification rate:

1. Check that the scanning parameters are appropriate, and that all the data in the form exists in the scanned image.

2. Check that the empty form image (EFI) was scanned properly.

3. Prepare a batch of forms for every EFI.

4. Create a multi-page TIF file with this batch.

Build interfaces to other software components

In this stage, you build the interface between eFlow files and other components. This stage varies from one customer to another. The following options are available:

- Create reports using the TIS XML export data component
- Transfer the exported XML/ASCII file to another medium

Chapter 2

Design window

To open **Design**, in the **eFlow LaunchPro**, from the **Tools** list, select **Design**.

The **Design** window appears.
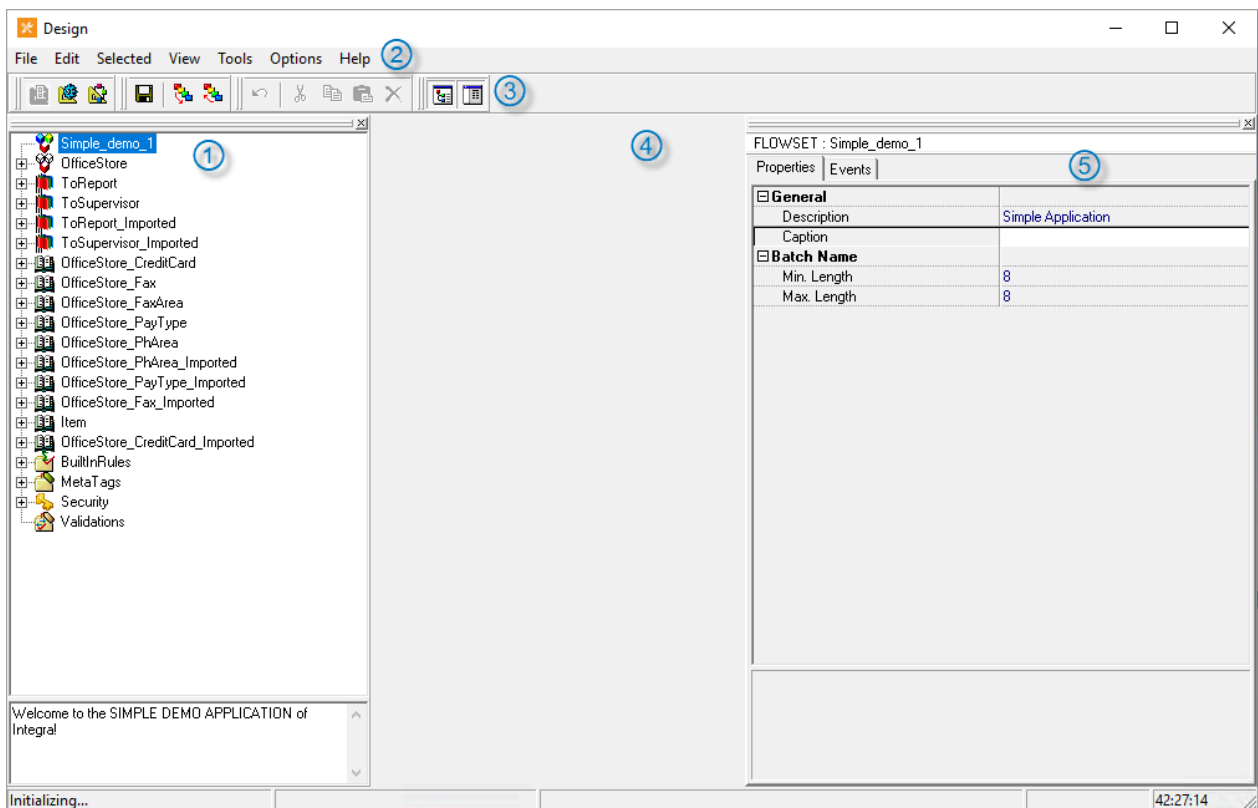
The Design window (Application designer) includes the following sections:

1	Application Explorer
2	Menu bar
3	Toolbar
4	Working area
5	Object Inspector

Application Explorer

All components of an application are displayed in a tree structure in the Application Explorer. The tree hierarchy displays open flows and all items contained in each flow. Click **Expand** ⊞ next to any tree node to view the information within it.
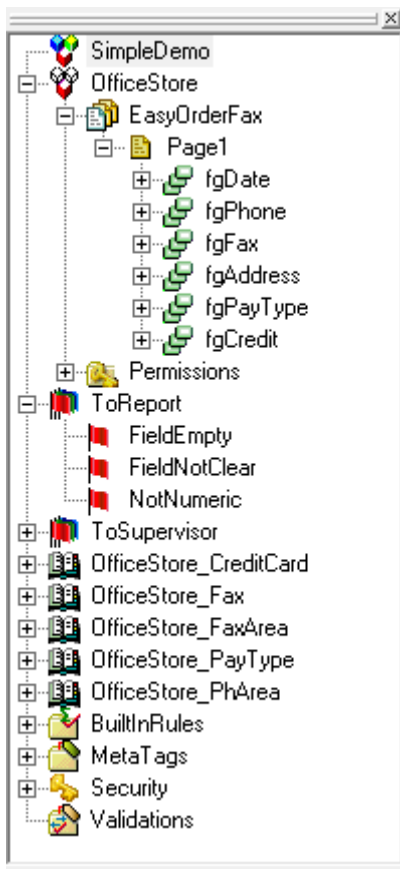
Object icons

An application contains many different objects, represented by the following icons:

Icon	Object	Refer to
	Application	Application design
	Flow	Design flow
	Form	Design Form and Page
	Page	Design Form and Page
	Field Group	Design field group and field
	Field	Design field group and field

Icon	Object	Refer to
	Table	Table design
	Column	Table design
	Rule	Define rules and validation functions
	Validation Function	Define rules and validation functions
	Exception Category	Define exceptions
	Exception	Define exceptions
	Lookup table	Define lookup tables
	Lookup item	Define lookup tables
	Permission	

Copy and paste an object

Objects can only be copied where logically appropriate, that is, to a level hierarchically superior to the object in the application tree. For example, a page may only be added to a form, a field may only be added to a field group, and so on.

1. Click to select the source object that you want to copy.

2. Right-click to open the context menu and select **Copy**.

3. Click to select the target object, to which you want to paste the source object.

4. Right-click to open the context menu and select **Paste**.

A copy of the source object appears in the application tree under the target object. You can change any properties as appropriate.

Cut an object

When you cut an object, you can still paste it back into the application object tree.

1. Click to select the object.

2. Right-click the object to open the context menu, and select **Cut**.

The object disappears from the application tree.

Delete an object

Once you have deleted an object, it is removed from the system permanently.

1. Click to select the object.

2. Right-click the object to open the context menu and select **Delete**.

The object disappears from the application tree.

Working Area

The Working Area is where you view images of empty forms. Using these empty form images, you can instruct eFlow how to process specific fields in a form, or what parts of the form should be used to identify it.

The Working Area toolbar contains items that relate to the object selected in the Application Explorer. When a form is selected, for example, the Working Area contains a thumbnail image of an empty form linked to the flow. You can also display a full-scale image of an empty form.

Using the Working Area toolbar, you can designate areas, or Regions of Interest (ROIs), where data recognition is to take place in the Recognize module. For detailed information on the Working Area toolbar and working with empty form images see Design EFI.
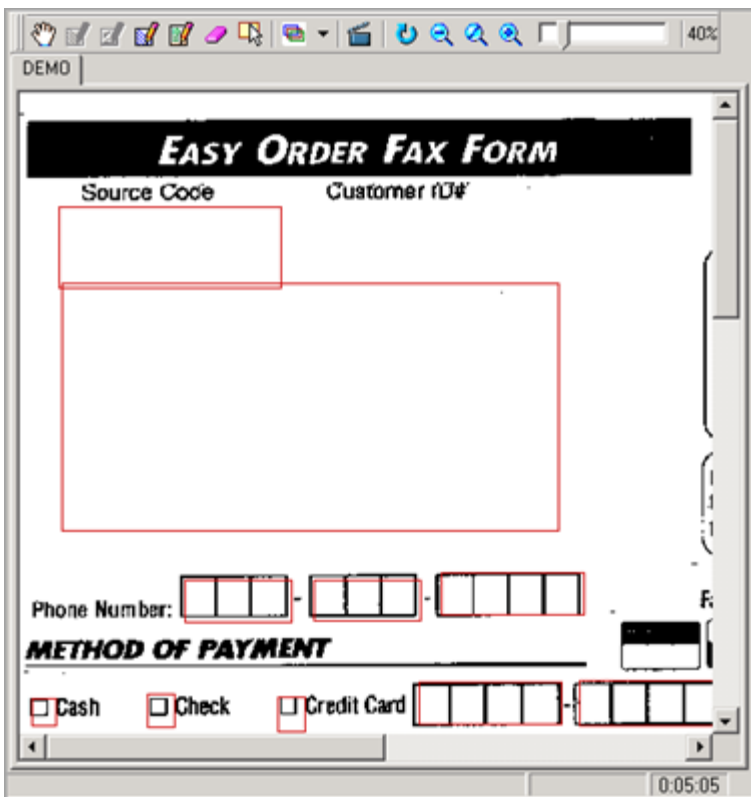
Object Inspector

The Object Inspector displays the properties and events of any object selected in the Application Explorer. In the Object Inspector, you change the values of properties or configure events.

By default, the **Properties** tab is always displayed in the Object Inspector. To view and edit events, click the **Events** tab.
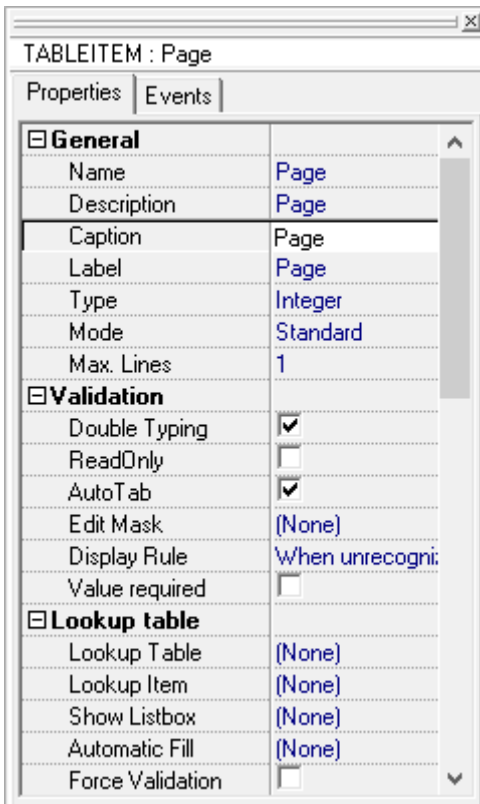
The properties of an object (flow, form, page, field group, field, table) define that object. Properties may be logical (name, type of data, format of data) or physical (size, location). For detailed information on object properties, see the Properties section under each type of object.

You can assign events to objects. Events are actions that are triggered at a given station when the object is entered, exited, or a hotkey is pressed. .NET or Win32 DLLs are attached to these events to run special actions. For detailed information on events, see Event functions for more information.

Toolbar

The buttons on the toolbar provide quick access to commands and options, which you can also access from the menu bar.

Position the mouse cursor over a button to display a tool tip that explains its purpose.

Button	Description
	Switch to the Application Designer.
	Switch to the OCR Engines Designer.

Button	Description
	Switch to the Workflow Designer.
	Save the entire application, including all the flows, forms, pages, exceptions and all other tree objects, and including all properties and events.
	Expand the application tree to see all the objects in the tree.
	Collapse the application tree to hide the objects in the tree, displaying only the application name, flow names and names of several categories, such as exceptions or lookup tables.
	Undo. Not implemented in this version.
	Cut an object.
	Copy an object.
	Paste a copied or cut object.
	View or hide the Application Explorer.
	View or hide the Object Inspector.

Menu bar

The menu bar provides access to the following menus:

- File
- Edit
- Selected
- View
- Tools
- Options
- Help

File menu

Menu item	Description
Save	Save the entire application, including all the flows, forms, pages, exceptions and all other tree objects, and including all properties and events.
Check Application Integrity	Check the design for usage errors and open a dialog box indicating problems. Problems refer to errors listed in the logger. It is especially useful for check boxes, if one of the values has been forgotten. If there are no problems, a No Integrity Problems message is displayed.
Restart Application	Restart the application.
Expand Application Tree	Expand the application tree to see all the objects in the tree.

Menu item	Description
Collapse Application Tree	Collapse the application tree to hide the objects in the tree, displaying only the application name, flow names and names of several categories, such as exceptions or lookup tables.
Exit	Exit the eFlow Design module.

Edit menu

Menu item	Description
Undo	Not implemented in this version.
Cut	Cut an object.
Copy	Copy an object.
Paste	Paste a copied or cut object.
Delete	Delete an object.

Selected menu

The **Selected** menu contains commands that are available for the selected (highlighted) object in the Application Explorer tree.

ℹ Some commands are only available for a specific object. For example, you can only create a page in a form. Other commands may be available for more than one object.

Object selected	Menu items shown	Description	Refer to
Application	New Flow	Create a new flow.	Create a flow
	New Exception Category	Create a new exceptions category (used to sort exceptions for workflow routing) in the application.	Create an exception category
	New Lookup Table	Create a new lookup table (a selection list of possible field values used in the Validate station) in the application.	Create a lookup table
	Import Flow	Import an existing flow from another application.	
Flow	New Supported Permission	Add and edit permissions for user/station access and actions.	
	New Form	Create a new form (physical form containing many types of data) in the flow.	Create a form

Object selected	Menu items shown	Description	Refer to
	Set EFI Order	Arrange the empty form images (EFIs) attached to the flow.	
Form	New Supported Permission	Add and edit permissions for user/station access and actions.	
	New Page	Create a new page in the form.	Create a page
	Edit Screen Layout	Open the Layout Designer to create a screen layout in the Validate station.	Screen layout design
	New VFunction	Create a validation function (checks on allowable data).	Create a validation function
Page	New Fields Group	Create a new field group (fields linked together in a specific sequence) in the page.	Create a field group
	Restore Tables	Restore previously created tables.	
Field Group	New Supported Permission	Add and edit permissions for user/station access and actions.	
	New Field	Create a new field (blank spaces where the form data is filled in) in the field group.	Create a field
	New Table	Create a new table for the field group.	Create a table
	Edit Screen Layout	Open the Layout Designer to create a screen layout in the Validate station.	Screen layout design
	New VFunction	Create a validation function (checks on allowable data).	Create a validation function
Table	New Column	Create a new column in the table.	Create a column
Field or Column	New Supported Permission	Add and edit permissions for user/station access and actions.	
	New Rule	Create a rule (logical relationships between fields and/or columns) for the object.	Create a rule

Object selected	Menu items shown	Description	Refer to
	New VFunction	Create a validation function (checks on allowable data).	Create a validation function
Permissions	New Supported Permission	Add and edit permissions for user/station access and actions.	
Exception Category	New Exception	Create a new exception (notification in the Validate station of invalid data in a field) for the exception category.	Create an exception
Lookup Table	New Lookup Item	Add a new lookup item to an existing lookup table.	Create a lookup table
BuiltInRules	Restore to default	Undo all changes to all built-in rules.	
Rule	Restore to default	Undo all changes to the built-in rule.	
MetaTags	New MetaTag	Create a new meta tag.	
Stations	New Supported Permission	Add and edit permissions for user/station access and actions.	
Services	New Supported Permission	Add and edit permissions for user/station access and actions.	
Roles	New role	Add and edit permissions for user/station access and actions.	
Validation	New validation category.	Create a new validation category.	

View menu

Menu item	Description
Object Inspector	View or hide the Object Inspector.
Tree View	View or hide the Application Explorer.
Working Area Tool Palette	View or hide the Working Area toolbar.
View Debug Information	View the application debug information (TIS Logger). The logger is a separate window of the application accessed through the system tray.

Tools menu

The **Tools** menu contains a list of commands that are available for the selected object in the application explorer object tree.

ℹ Some commands are only available for a specific object. For example, you can only create a page in a form. Other commands may be available for more than one object.

Menu item	Submenu option	Description	Refer to
Get New EFI From File		Obtain an EFI from a selected image file.	Create an EFI from an image file
Scan New EFI		Obtain an EFI by scanning in the form.	Scan an EFI from a paper form
Copy EFI		Copy the selected EFI.	Copy an EFI
Delete EFI		Delete the selected EFI.	Delete an EFI
Edit Screen Layout		Open the Layout Designer to create a screen layout in the Validate station.	Screen layout design
Run PlugIn	Application Designer	Switch to the Application Designer.	
	OCR Engines Designer	Switch to the OCR Engines Designer.	
	Workflow Designer	Switch to the Workflow Designer.	
Attachments Manager		View files (EFIs, lookup files, workflows) saved in the current application.	

Options menu

Menu item	Description
Check integrity on exit	If selected, the Design module checks that the application design operates correctly. Any errors are displayed in the logger.
Display node label	If selected, displays the **Caption** property instead of the **Name** property in the Application Designer.

Help menu

Menu item	Description
Homepage	Opens eFlow product documentation website.

Chapter 3

Building an application

To begin working with eFlow, you must create and define an application, which is essentially a workflow including one or more flows and the definition of all forms, fields, tables, rules, validation functions, exceptions, and so on, used in the application.

In an application, you define how many flows and forms you want, which EFIs you want to attach to them, and how you want the data processed. You define which OCR engines will be used to interpret the form entries, and what type of image enhancement and form removal parameters to apply to achieve the best results. You can group data items and apply logical and mathematical rules to ensure data correctness and quality, and to reduce the number of exceptions that are returned to the operator for manual data completion.

You can define many applications in the same eFlow installation. Each application contains its own flows, forms, users, and exceptions. There is usually one application per project.

> ⓘ You can create and run more than one application, but you can only work on one application at a time in the Application Designer.

Create an application in the Administrate module

eFlow applications are configured in the eFlow Design module but cannot be created or activated there. To create a new application, or to make an existing application available in eFlow Launch and eFlow Launch Pro, you must add the application in the Administrate module.

To create an application, refer to the "Add an application" section in *Kofax eFlow Administrate User Guide*.

Build the application in the Design module

Once an application is created in Administrate, build the application in the Design module and perform the following tasks:

1. In the Design module, in the Application Explorer, click on the application.

2. Configure the application properties.

3. Add objects to the application:

   a. Flow

   b. Forms and pages

    c. Field groups and fields

    d. Tables

    e. EFIs

    f. Define exceptions

    g. Define lookup tables

    h. Define rules and validation functions

    i. Design screen layout

    j. Define OCR engines used by the application

    k. Define application workflow

    l. Define virtual engines

Application properties

In the Design module, click on the application in the Application Explorer tree and edit the properties in the Object Inspector.

General properties

Property	Description
Description	A short description of the application.
Caption	A name for the application that appears throughout the system.

Batch Name properties

Property	Description
Min. Length	Minimum length for the name of batches in the application.
Max. Length	Maximum length for the name of batches in the application.

Chapter 4

Design flow

A flow is a collection of definitions describing processing, enhancement and other parameters that define how batches of forms are identified and processed in the eFlow system.

An application may consist of one or more flows. A flow represents a given path through a workflow. Similar forms may require different processing, some passing through more, fewer, or different stations than others. In this case, you create two or more flows within an application so that different forms are processed according to their differences in format and content.

Physical forms are grouped into batches, or collections, before they are scanned. Then, every batch is put into the scanner. The scanner scans each page in sequence and creates an image file containing all the pages of that batch. The scanning operator assigns the batch to a flow. Every batch is associated with a certain flow in the system.

The definition of the flow includes the following:

* The forms that are associated with the flow.
* The EFIs that match a form in the flow.
* The properties that are associated with the runtime needs and stages of this flow (data collection, content recognition, content validation and content delivery).

Create a flow

Flows can only be added to the application object.

1. In the Application Explorer tree, click the application object to select it.
2. Right-click the highlighted object to open the context menu and select **New Flow**.
   The flow object appears in the Application Explorer tree. The Object Inspector displays the properties and events.
3. Type a name and press Enter.
4. Set the flow properties.

General properties

Property	Description
Name	The name of the flow. This name appears automatically in the **Description** box in the **Object Inspector** window after you create a flow. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the flow.
Caption	A name for the flow that appears on the screen throughout the application.
Image Resolution	The resolution of the scanned EFI. The default setting is 200 dpi.

Input properties

In the input flow properties, define information about the input files.

Property	Description
Input Form	You may want to associate information with batches that varies from batch to batch. This is accomplished with a form that contains the fields you need, defined in eFlow terms as an "input form". You define an input form using the **Input Form** property. When you define a form as an input form, the scanner operator scans and assigns this form once for each batch. Input forms are only used in the input and export stations. In some cases, you may want to add a pop-up window for post-scan information (post-input). This allows the scanner operator to add information about batches. This information is contained in a logical form (without template). It is only used in the input stations. Information can include scan date, number of forms, number of pages, and so on. It can be used for validation purposes, for example, to ensure that the correct number of pages were in fact scanned.
Keep Batch Name	Usually, eFlow assigns its own names to the incoming batches by increasing the batch count number by one. Selecting the **Keep Batch Name** option allows the system to retain the batch name after scanning. Preserving the batch names helps in cases, for example, when multi-page TIF file names are given by the scanner.
Batch Name Prefix	Specify a prefix that the system will assign to any batch in the flow after scanning. Keeping a prefix with batch names helps in identifying the similar batches after scanning. ⓘ The Keep Batch Name and Batch Name prefix options help in preserving the integrity of your file naming convention.

Processing properties

The flow's processing properties relate to form recognition. They provide the form recognition engine with hints that can improve recognition. Correct form recognition is crucial, and eFlow provides many tools that help specifically identify forms using sophisticated technology.

Processing involves the following stages:

- Form image enhancement
- Form identification and removal
- Field image enhancement
- Field segmentation
- Optical character recognition (OCR) and/or intelligent character recognition (ICR)
- Voting for optimal results
- Storing the image as defined in the setup (TIF or DIF)

Property	Description
Default EFI	Select the default EFI that will be used if form identification fails on a specific page. EFIs are images of the empty forms. For every page in the form that is linked to the flow, you can define one or more EFIs. In cases where a form has several variations, you should assign several EFIs to the same form, one for each possible variation. When a form cannot be matched to any of the EFIs that have been defined for the forms in the flow, the default EFI specified here is used. See Design EFI for more information about EFIs. ⓘ You can only choose from one of the EFIs that have been linked to the flow's pages. If you have not yet created and linked any EFIs, no EFIs will be available. You must first create the EFI. See Create an EFI from an image file. For more information about linking EFIs to the flow's pages, see Link an EFI to a page.
Unrecognized Char	Click ⋯ and select the symbol that represents an unrecognized character. The operator in the Validate station sees the selected character (such as *) in place of unrecognized characters.

Enhancement properties

Enhancement enables post-scan image modification. You can use many enhancement parameters, such as:

- **Deskew** to correct alignment
- **Noise removal** to remove different kinds of noise, such as dots and specks
- **Lines** to remove vertical or horizontal lines
- **Character enhancement** to improve character recognition
- **Rotation** to correct an image that was rotated when it was scanned

You can perform enhancement in more than one place in the flow. In each instance, you can combine elements of the different enhancement filters.

You can use the The Enhancement Builder to define sets of image enhancement filters that are applied in the Recognize module. The following options are available.

Property	Description
Before FormID	Click ⸱⸱⸱ to define the enhancement filters that are applied to each page before the page is identified. The FormID algorithm allows the Recognize module to recognize the type of form and process it accordingly. One or more regions in the EFI are defined as form recognition regions, in the case where there are several types of forms that differ only by these regions. If at least one FormID region does not comply with predefined conditions, the form is not matched to the EFI. See Define page level regions for more information. In certain cases, the quality of the form image may make recognition difficult. Setting enhancement filters can help to improve form recognition. See The Enhancement Builder for more information.
Different For Odd/Even Pages	Select this check box to set two different sets of enhancement filters, one set for even pages and another set for odd pages. ⊟ **Enhancement** Before Form ID (Odd Pages) — (None) ⸱⸱⸱ Before Form ID (Even Pages) — (None) Different For Odd/Even Pages — ✔
On Identified Page	Click ⸱⸱⸱ to define the enhancement filters that are applied to each recognized page before the page is saved as a DIF image. A DIF file contains the scanned image after form removal.
On Unidentified Page	Click ⸱⸱⸱ to define the enhancement filters that are applied to pages that have not been recognized. The enhancement is applied if the page is not matched.

The Enhancement Builder

The Enhancement Builder enables you to define image enhancement filters and immediately test the results using a supplied sample file. You can also test the image enhancement operations on your own images, assuming they are stored as files.

To open the Enhancement Builder, click ⸱⸱⸱ to the right of an enhancement property. The **Setup Image Enhancement** dialog opens.
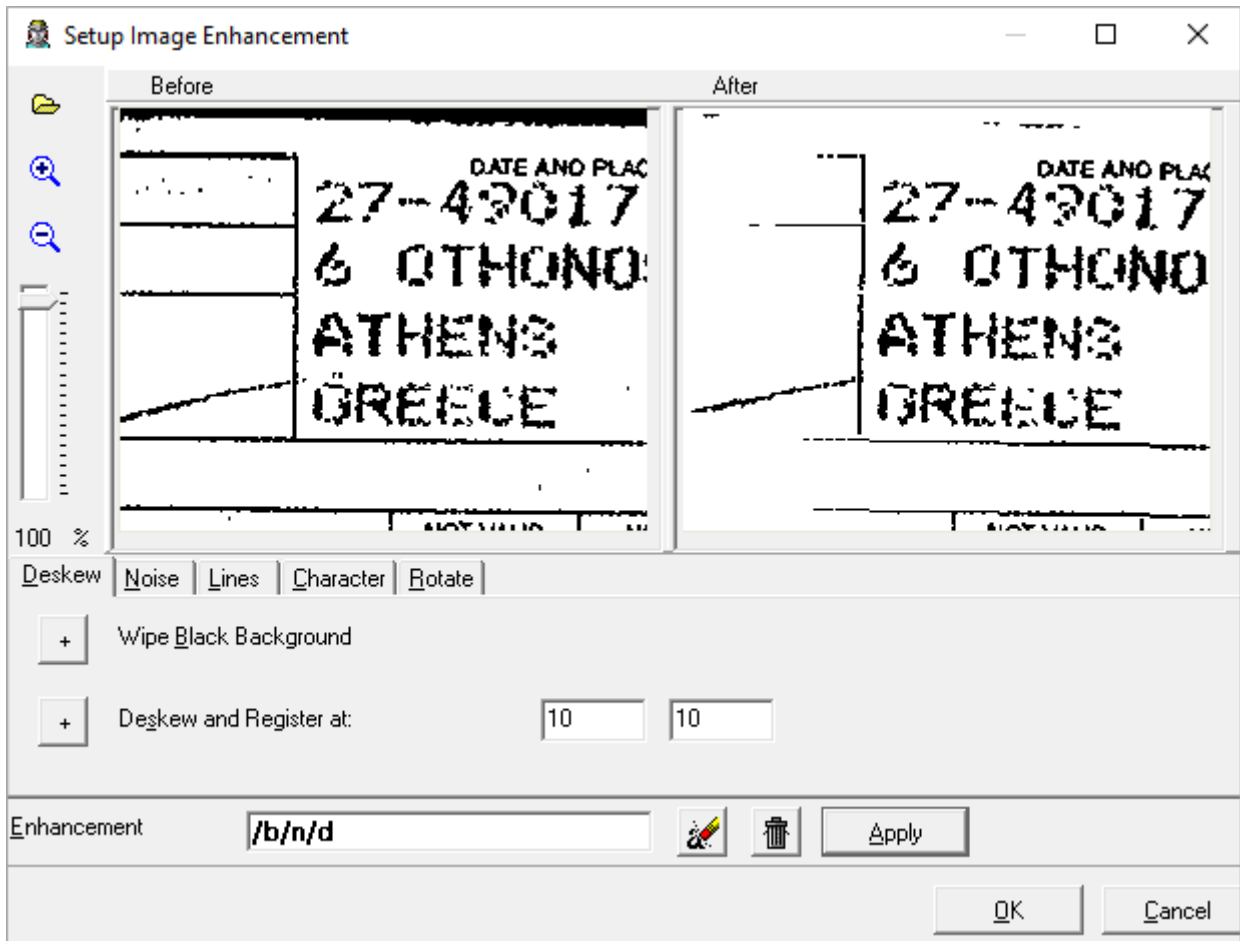
Image enhancement comprises one or more customized enhancement filters:
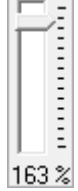
- **Deskew** to correct alignment
- **Noise removal** to remove different kinds of noise, such as dots and specks
- **Lines** to remove vertical or horizontal lines
- **Character enhancement** to improve character recognition
- **Rotation** to correct an image that was rotated when it was scanned

Make sure that the selected filters do not corrupt the image such that they interrupt the work of the OCR. You must use a trial and error procedure to test the filters. Use FormOut! to test page recognition (see Test an EFI).

The image on the left is the image before enhancement, while the image on the right is the enhanced image obtained using the defined filters.

Use the following buttons to change the display.

Button	Description
🖿	Select another image file.

Button	Description
(magnify icon)	Magnify the displayed image. Each click magnifies by 25%.
(shrink icon)	Shrink the displayed image. Each click shrinks by 25%.
(slider 163 %)	Shrink or magnify the displayed image in increments. Use the slider or click in the bar. Each click shrinks or enlarges the image in increments of 100%.

As you add enhancement filters from the available options, the corresponding commands, called "tokens", are added to the string in the **Enhancement** box. For example, if you click the **Wipe Black Background** button, the token /b is displayed in the **Enhancement** box. If you click **Deskew and Register at** and define the desired coordinates, the token **/d10,10** is displayed.
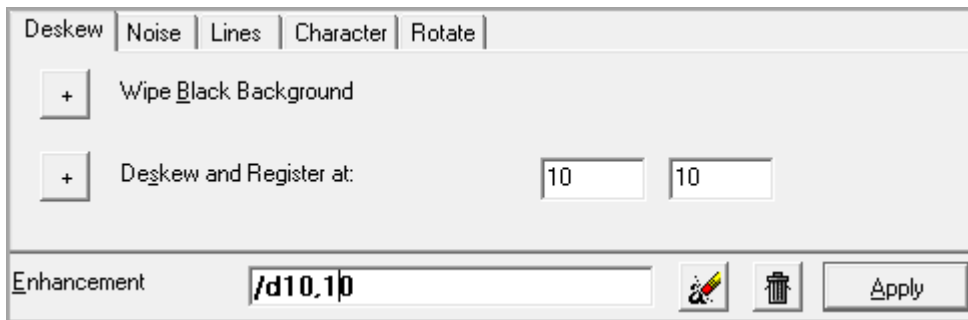
The following buttons affecting these tokens appear in all tabs:

Button	Description
(delete last token icon)	Delete the last token applied.
(trash icon)	Delete all tokens from the **Enhancement** box.
Apply	Apply the selected enhancement and update the enhanced image viewing area accordingly.

ⓘ To change the values, first remove unwanted tokens from the enhancement string. Either delete the tokens directly in the **Enhancement** box, or click the **Delete last token** button to remove tokens from the end. Changing the properties and clicking **Apply** will not apply new properties.

Correct skewed images and black backgrounds

In the **Deskew** tab, you can do the following:

- Erase the black background that is produced by some scanners around the page.
- Correct a skewed image and apply registration (alignment) according to specified coordinates.
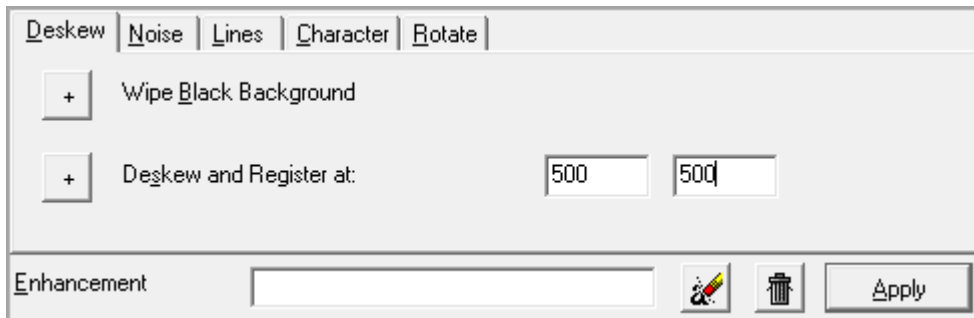
The options in this tab apply to whole pages rather than to single ROIs.

Erase a black background

1. Click ⊞ for **Wipe Black Background**.
   The corresponding command token is added to the enhancement string.

2. Click **Apply** to view the effect on the image.
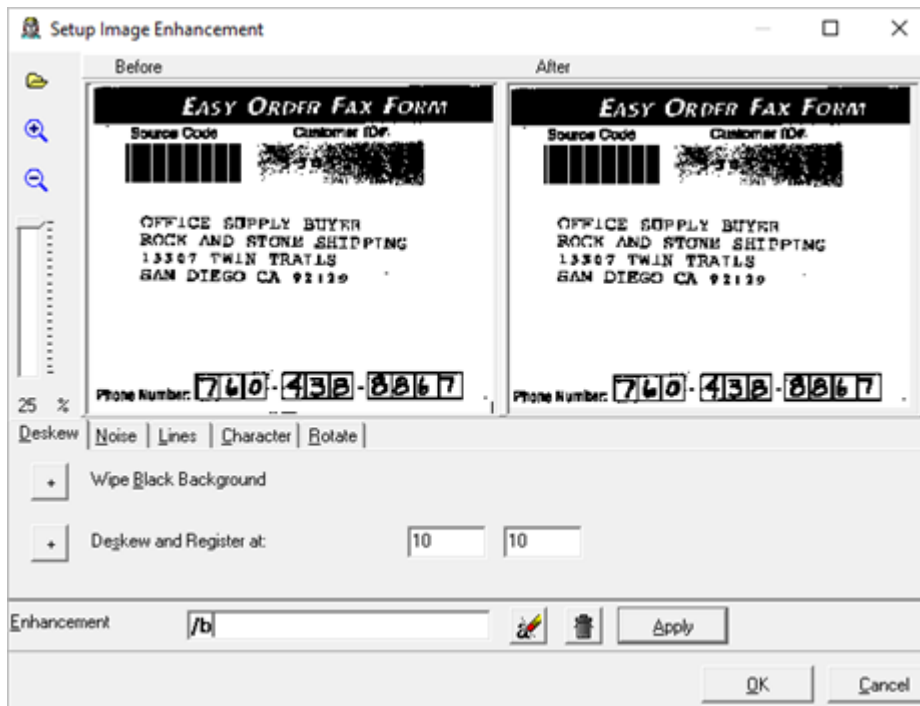
Correct a skewed image

To deskew or realign an image, you must specify the new coordinates of the top left-hand corner of the image. This is called the registration point.

1. In **Deskew and Register at** boxes, enter the pixel coordinates (**x,y**) of the new registration point.

2. Click ⊞ for **Deskew and Register at**.

   The corresponding token and selected values are added to the enhancement string.

3. Click **Apply** to view the effect on the image.

ℹ️ In the **Noise**, **Lines**, and **Character** tabs (whose descriptions follow), enhancement properties defined via the **EFI Setup** window will be applied to one ROI only. To apply these properties to the whole page, you must define them from the **Process** tab in the **Flow Setup** window.

Eliminate noise and shadow

In the **Noise** tab, you can eliminate excess noise (specks) and shadows from a scanned image.

ℹ️ Use this enhancement feature with care, as it increases processing time.
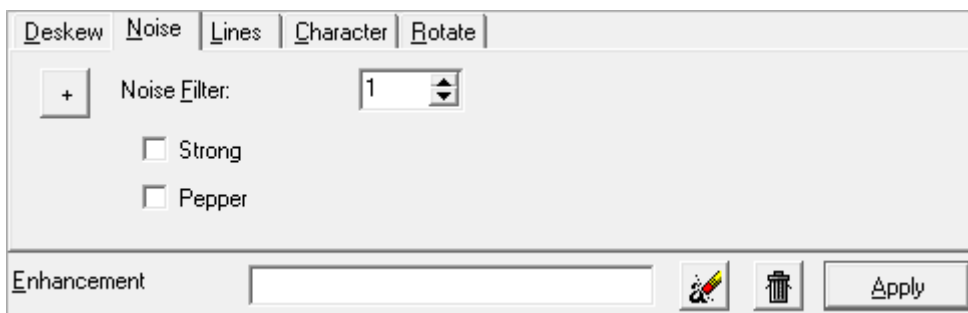
With the **Noise Filter** options, you can define the maximum size of a speck of noise (n by n pixels) and the method of noise removal.

1. In the **Noise Filter** box, enter the maximum speck size, in pixels.

2. To apply a more thorough form of noise removal, select the **Strong** check box.

3. To apply the pepper algorithm for noise removal, select the **Pepper** check box.
   Strong and **Pepper** are two specific algorithms for noise removal.

4. Click ⊡ for **Noise Filter**.

5. Click **Apply** to view the effect on the image.

Eliminate lines

In the **Lines** tab, you can eliminate horizontal and vertical lines that are part of the EFI, leaving all the characters intact.

The default is to remove only those lines that are longer than 50 pixels, and whose length is at least 20 times greater than the thickness.

1. In the **Removal Horizontal Lines** box, enter the minimum line length for removal of horizontal lines.

2. In the **X/Y Ratio** box, enter the minimum thickness ratio for horizontal lines.

3. In the **Removal Vertical Lines** box, enter the minimum line length for removal of vertical lines.

4. In the **Y/X Ratio** box, enter the minimum thickness ratio for vertical lines.

5. Click ⊡ for **Remove Horizontal Lines** and **Removal Vertical Lines**.
   The corresponding token and selected values are added to the enhancement string.

6. Click **Apply** to apply the filters and view the effect on the image.

Define character enhancements

In the **Character** tab, you can define how characters are enhanced. You can define values to compensate for a low-quality printer (such as a dot-matrix printer), and to connect characters that have been broken during scanning.

> 🛑 Use this enhancement feature with care, as it greatly increases processing time. It should not be applied to a whole page.
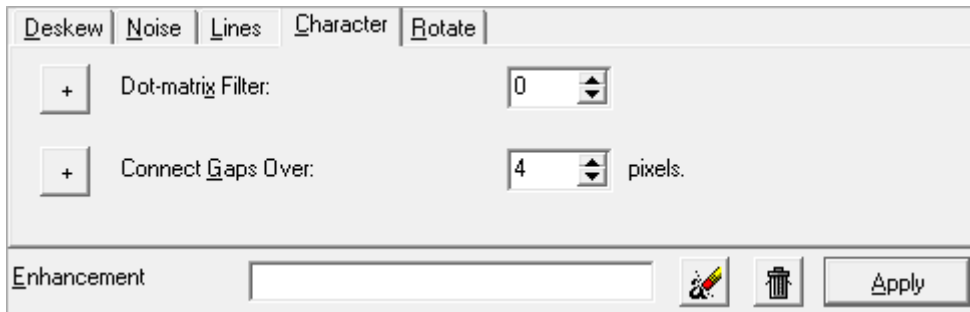
1. For **Dot-matrix Filter**, enter a value in the box and click ⊞.
   The corresponding token and selected values are added to the enhancement string.

2. For **Connect Gaps Over**, enter a value in the pixels box and click ⊞.
   This value defines the size of the gaps that will be connected.

   > ℹ️ Note: Setting too high a value may add extraneous lines.

   The corresponding token and selected values are added to the enhancement string.

3. Click **Apply** to apply the filters and view the effect on the image.

Rotate images

In the **Rotate** tab, you can rotate the image in increments of 90°.
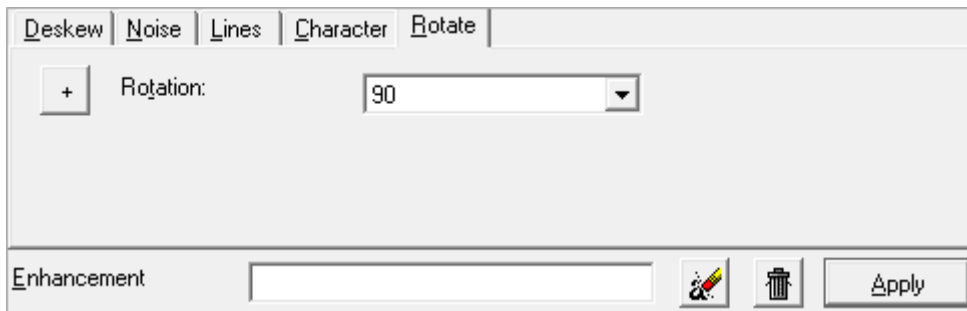
1. For **Rotation**, select the image rotation from the drop-down list and click ⊞.

2. Click **Apply** to rotate the image.

Export properties

In Export properties, you define which images or data are saved, and where they are saved.

Property	Description
Original Input	Select this check box to export the original images, including JPG and GIF files.

Property	Description
Original TIFF	Select this check box to export the original input TIF.
Data Format	Select the format of the exported data: • **INI file.**: If selected, data is exported in ini format. • **Delimiter Separated**: If selected, a new property appears, where you select the delimiter character. • **XML file**: If selected, data is exported in XML format.
Exception Report Format	Select the format of the exception report: • **INI file.**: If selected, data is exported in ini format. • **Delimiter Separated**: If selected, a new property appears, where you select the delimiter character. • **XML file**: If selected, data is exported in XML format.
Default Encoding	Select the character encoding that will be used by the XML document, if files are exported as XML.
Export Path	Specify the path to which the images will be exported.

Chapter 5

Design form and page

A physical form is the original input – a scanned paper document – that eFlow converts to electronic data as output. A physical form may contain many types of data, such as logos, barcodes, name, date, address, inventory tables, census data, check boxes, and many other types of numerical and text data.

A logical form is the eFlow platform counterpart to a physical form. The logical form is a structure that defines the data captured during the data entry process.

After creating or opening a flow, the next step is to create a new form and a new page. A flow can have one or more forms, with each form containing one or more pages.

Create a form

Forms may only be added to a flow.

1. In the Application Explorer tree, click to select the flow object.
2. Right-click the highlighted object, to open the context menu and select **New Form**.
   The form object appears in the application object tree.
3. Type a name and press Enter.
   The Object Inspector displays the properties and events.
4. Set the form properties.

Form properties

Property	Description
Name	The name of the form. The name should be alphanumeric. This name appears automatically in the **Description** box in the Object Inspector window after you create a form. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the form.
Caption	A name for the form that appears on the screen throughout the application.

Create a page

Pages may only be added to a form.

1. In the Application Explorer tree, click to select the form object.
2. Right-click the form object to open the context menu and select **New Page**.

   The page appears in the application object tree. The Object Inspector displays the properties and events.
3. Type a name and press Enter.
4. Set the page properties and link an EFI to the page.

Page properties

Property	Description
Name	The name of the page. The name should be alphanumeric. This name appears automatically in the **Description** box in the Object Inspector window after you create a page. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the page.
Caption	A name for the page that appears on the screen throughout the application.
EFI	You must link an EFI to each page of the form. An EFI is like a template of the structured document, which the system uses to extract the data.
	You must create the EFIs before you can link them. See Create an EFI for more information.
	Click ··· to select an EFI. See Link an EFI to a page for more information.

Link an EFI to a page

For every form that is linked to the flow, and for every page in the form, you can define one or more EFIs. EFIs are images of the forms without the filled-in data. An EFI is like a template of the structured document, which the system uses to extract the data. In many cases, a form has several variations, so you can assign several EFIs to the same form or page. See Design EFI for more information.

You must link an EFI to each page of the form. You must create the EFIs before you can link them. See Create an EFI for more information.

1. In the Application Explorer tree, click to select the page object.
2. In the Object Inspector, click ··· to the right of the **EFI** property.

   The **EFIs Linking** dialog box opens. In the application tree, under each page is a list of EFIs with check boxes.

3. To view an EFI, click on the name of the EFI (not the check box).

   The selected EFI is displayed in the preview panel on the right.
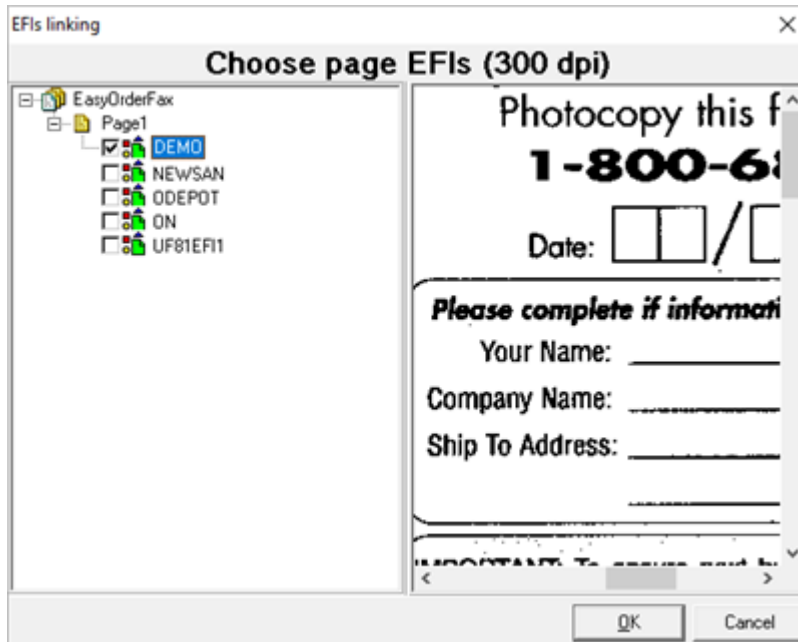
4. Select the check box next to an EFI to link it to the page. You can link more than one EFI to a page.

5. Click **OK**.

To remove the link between an EFI and the page, clear the check box next to the EFI.

Order pages

The pages appear in the Application Explorer tree in the order in which they are displayed to the operator in the Validate station. The physical display layout of the pages may differ, but the operator in Validate will move from one page to the next in the order specified here (by pressing the Enter key to proceed from field to field).

To rearrange pages, click to select a page, drag it with the mouse to the desired location, and drop it.

All the page's properties and any sub-contents (field groups, fields, tables, and so on) move with it.

Chapter 6

Design field group and field

Each form is a set of fields - the "blanks" where the variable data, such as a name, an address, or a date, is to be filled in. This variable data can be handwritten, typed, OMR elements such as check boxes, or bar-coded information. The field values to be processed are extracted from ROIs (Regions of Interest) on the image and are sent for processing. ROIs are areas of the image that contain data for which OCR will be performed.

Fields have a dual purpose:
* They are used to map ROIs in the page image to eFlow data fields.
* They are the input fields that the operator sees when keying in data in the Validate station.

Field groups allows fields to be grouped together and linked in a specific sequence for event management and easy viewing in the Validate station. Field groups display the selected fields together in a particular order in the Validate station to facilitate data entry. It is therefore important to pay attention to order when adding fields to a field group. The field order as it appears in the Application Explorer tree will be reflected exactly in the field order of the Validate station.

> ⓘ Field groups can contain fields that appear in separate areas of the form. The fields in the group do not have to be adjacent.

Create a field group

Field groups can only be added to a page.

1. In the Application Explorer tree, click to select the page object.
2. Right-click the page object to open the context menu and select **New Fields Group**.
   The field group appears in the application object tree. The Object Inspector displays the properties and events.
3. Type a name and press Enter.
4. Set the field group properties.

Field group properties

Property	Description
Name	The name of the field group. The name should be alphanumeric. This name appears automatically in the **Description** box in the Object Inspector window after you create a field group. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the field group.
Caption	A name for the field group that appears on the screen throughout the application.

Order field groups

The order in which the field groups appear in the Application Explorer tree is the order in which they are displayed to the operator in the Validate station. The physical display layout of the field groups may differ, but the operator in Validate will move from one field group to the next in the order specified here (by pressing the Enter key to proceed from field to field).

To rearrange field groups, click to select a field group, drag it with the mouse to the desired location, and drop it.

All the field group's properties and any sub-contents (fields, tables, and so on) move with it.

Field group events

You can assign events to a field group in the **Events** tab of the Object Inspector. Events are actions that are triggered at a given station when the object is entered, exited, or a hotkey is pressed. .NET or Win32 DLLs are attached to these events to run special actions.

Typical events may be inserting a necessary character not included in the form data, deleting an unnecessary character included in the form data, changing a common, known incorrect character to a known correct one, and so on.

See Event functions for information on the events available for field groups and how to assign events.

Create a field

Fields can only be added to a field group.

1. In the Application Explorer tree, click to select the field group object.
2. Right-click the field group object to open the context menu and select **New Field**.

The field appears in the application object tree. The Object Inspector displays the properties and events.

3. Type a name and press Enter.

4. Set the field properties.

Field properties

The field properties are grouped under the following sections:
- General properties
- Validation properties
- Lookup table properties
- Tile properties
- Bad Character Level properties
- Freedom properties
- Other properties

General properties

Property	Description
Name	The name of the field. The name should be alphanumeric. This name appears automatically in the Description box in the Object Inspector window after you create a field. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the field.
Caption	A name for the field that appears on the screen throughout the application.
Label	The label that is displayed next to the field in manual stations.
Type	Specify the type of data permitted in the field. See Field Type settings for more information.
Mode	Specify the display mode in the Validate station for regular fields and special fields in electronic forms. • **Standard**: A field that can be filled in. • **Radio Group**: Displayed as radio buttons. Only one option can be selected. (Only for electronic forms). • **Check Group**: Displayed as check boxes. More than one option can be selected. (Only for electronic forms).
Auto Tab	Select this check box to automatically move the cursor to the next field when the Validate station operator enters the maximum number of characters in the field.
Max Lines	Maximum number of permitted lines in the field.

Field Type settings

You can specify the type of data permitted in a field.

Click ⋯ to the right of the **Type** property to display the **Field Type Settings** dialog box.

In this dialog box, you can create and edit "mask" formats. A mask represents the text that is valid in a field. It also restricts the characters that a user can enter in the field in the Validate and Escalate stations. The validity of entered characters and formats are checked, and if the user attempts to enter an invalid character, the character will not be accepted. For example, in a data entry field for telephone numbers, you might define a field type that accepts only numeric input. If a user tries to enter a letter in this box, the application will not accept it.

Click on the corresponding button to select a predefined mask: **Text**, **Integer**, **Float**, **Date** or **Time**.

The mask format appears in the boxes below the buttons. You can modify the mask or use it as it is. To erase a mask, click **Erase Mask**.

For more information on editing masks, see Edit Mask settings.

Type	Description
Text	Only alphabetic text is permitted. You can specify the minimum and maximum permitted length of the text.

Type	Description
Integer	Only digits are permitted (without decimal point). You can specify the minimum and maximum permitted length of the number. Integers have a maximum length of 9 digits. • Min: The value of the field cannot be less than this value. If you do not assign a value, the minimum value is not examined. • Max: The value of the field cannot exceed this value. If you do not assign a value, the maximum value is not examined. • Default: If there is no other value, this value is applied to the field.
Float	Only digits are permitted (with decimal point). Settings for a float type are like those of an integer type. You can specify the minimum and maximum permitted length of the field value, and how many digits are expected before and after the decimal point. For **Min** and **Max** length, the decimal point does not count, for example, 1.23 = a length of 3 digits. **Decimal Digits** is the number of digits after the decimal point. If you want only integers, select zero. (This is the same as selecting the **Integer** type.) The floating digit **Decimal Length** is always the floating digit + 1. You can also specify a default value for the field.

Type	Description
Date	Only date formats are permitted. The following formats are available (d=day, m=month, y=year): • dd/mm/yy (default) • mm/dd/yy • yy/mm/dd • dd/mm/yyyy • mm/dd/yyyy • yyyy/mm/dd Predefined date fields are fixed and cannot be changed, that is, only the date formats listed above are available from this list. If you wish to specify another date format, you must create a mask property. In the **Min** and **Max** fields, you can specify a minimum and maximum acceptable date for the field. Enter the dates in the format that matches the date format you selected. eFlow will accept only dates that fall within the specified range. When you select **Now** check boxes, eFlow takes the current date from the system as the minimum or maximum limit. For example, if you want the minimum acceptable date to be one week ago and the maximum acceptable date to be the current date, in the **Min** field, enter the date in the appropriate format 7 days prior to the current date (that is, one week ago). For the **Max** field, click the **Now** check box and leave the **Max** field blank (that is, today by default). eFlow will now accept only dates that fall between one week before the current date and the current date.
Time	Only time formats are permitted. The predefined time format is hh:mm. The predefined time field is fixed and cannot be changed. If you wish to specify another time format, you must create a mask property.

Validation properties

Specify what sort of validation checks will be performed on the data found in the fields.

Validation	Description
Double Typing	Select this check box to specify that completion results will be accepted after being confirmed at the Validate station. (That is, two operators must confirm results.)
ReadOnly	Select this check box to specify that the field is read-only and cannot be edited.
Edit Mask	Specify a mask to define which characters are acceptable in the field, and how the field will be displayed in the Validate and Escalate stations. See Edit Mask settings for more information.

Validation	Description
Display Rule	Specify under what conditions a field will be displayed in the Validate and Escalate stations. • **Always**: Always display the field. • **Never**: Never display the field. • **When empty**: Only display the field if it does not contain a value. • **When not empty**: Only display the field if it contains a value. • **When unrecognized**: Only display the field if its contents were not recognized.
Value Required	Select this check box to specify that the field must contain a value to be considered valid.

Edit Mask settings

With the **Edit Mask** setting, you can define which characters are acceptable in a field, and how the field will be displayed in the Validate and Escalate stations.

A mask represents the text that is valid in a field. It also restricts the characters that a user can enter in the field in the Validate and Escalate stations. The validity of entered characters and formats are checked, and if the user attempts to enter an invalid character, the character will not be accepted. For example, in a data entry field for telephone numbers, you might define a field type that accepts only numeric input. If a user tries to enter a letter in this box, the application will not accept it.
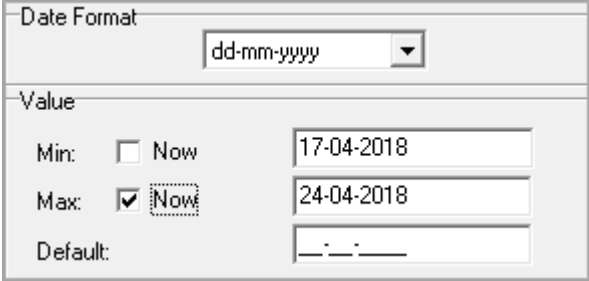
The **Edit Mask** setting relates to the field **Type** selected in the general properties.

Click the selection button ▥ to the right of the **Edit Mask** property to display the **Input Mask Editor** dialog box.
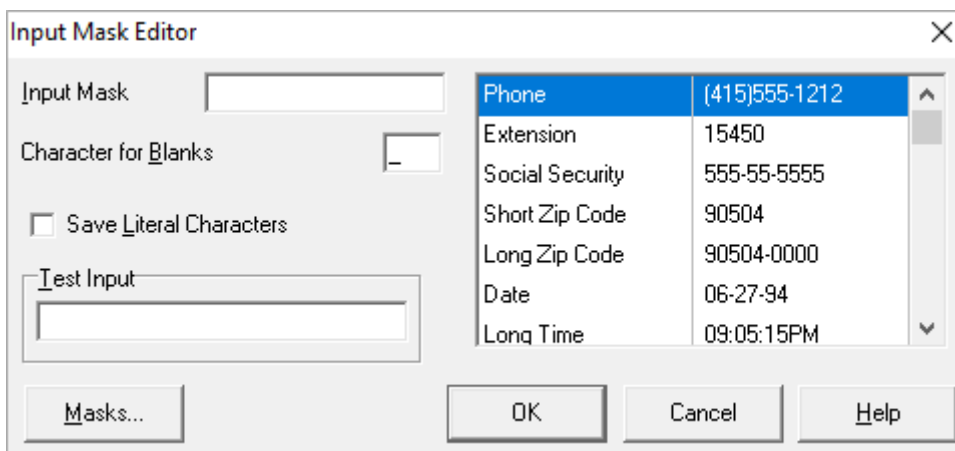
An input mask consists of three different sections separated by semicolons.

• The first section defines the length and character type of the data that may appear in the field, using the special characters defined in the table below.
• The second section is a single character that indicates whether literal characters from the mask should be included as part of the text.

- The third section is the character that is selected to fill in for blanks (that is, the character that replaces an empty space displayed in the field).

  For example, the input mask for a telephone number with area code could be the following string:

  (000)_000-0000;0;*

- The ten 0 characters in the first section specify that only a numeric character may appear in this position. The underscore character automatically inserts a space into the text.

- The 0 character in the second section indicates that literal characters (in this case, the parentheses) will not be included in the box. A text of 10 digits will be edited in the Validate and Escalate stations.

- The * character in the third section indicates that the asterisk is used as a blank in the mask.

The following special characters are used in the first section of the input mask to define the type of character that may appear in the field.

Any character that does not appear in this table may appear in the first section of the mask as a literal character. Special mask characters (those that appear in this table) may appear as literal characters if preceded by a backslash character (\).

Character	Description
!	If this character appears in the mask, optional characters are represented in the input mask as leading blanks. If a ! character is not present, optional characters are represented in the input mask as trailing blanks.
>	If this character appears in the mask, all characters that follow are in uppercase until the end of the mask or until a < character is encountered.
<	If this character appears in the mask, all characters that follow are in lowercase until the end of the mask or until a > character is encountered.
<>	If these two characters appear together in a mask, no case checking is done and the data is formatted with the case the user uses to enter the data.
\	The character that follows a \ character is a literal character. Use this character to use any of the mask special characters as a literal in the data.
L	This character requires an alphabetic character only in this position.
l	This (lowercase L) character permits only an alphabetic character in this position, but does not require it.
A	This character requires an alphanumeric character only in this position.
a	This character permits an alphanumeric character in this position, but does not require it.
C	This character requires an arbitrary character in this position.
c	This character permits an arbitrary character in this position, but does not require it.
0	This character requires a numeric character only in this position.
9	This character permits a numeric character in this position, but does not require it.
#	This character permits a numeric character or a plus or minus sign in this position, but does not require it.

Character	Description
:	This character is used to separate hours, minutes, and seconds in time fields. If the character that separates hours, minutes, and seconds is different in the regional settings of the Control Panel utility on your computer system, that character is used instead.
/	This character is used to separate months, days, and years in date fields. If the character that separates months, days, and years is different in the regional settings of the Control Panel utility on your computer system, that character is used instead.
;	This character is used to separate the three fields of the mask.
_	This character automatically inserts spaces into the text. When the user enters characters in the field, the cursor skips the _ character.

ⓘ When working with multi-byte character sets, such as Japanese shift-JIS, each special mask character represents a single byte. To specify double-byte characters using the L, l, A, a, C, or c specifiers, the mask characters must be doubled as well. For example, LL would represent two single-byte alphabetic characters or a one double-byte character. Only single-byte literal characters are supported.

You can either select the most appropriate mask from the list of predefined masks on the right side of the dialog box, or create your own mask using the parameters on the left side of the dialog box.

Use a predefined mask

Use the predefined masks scroll box on the right to scroll through the predefined masks. Click to select a predefined mask, the fields on the right are filled automatically.

The following predefined masks are available.

Mask	Description
Phone	Phone number with area code
Extension	Phone extension number
Social Security	Social security number
Short Zip Code	Zip code number
Long Zip Code	Zip code number with extra four digits
Date	Date in the format dd/mm/yy
Long Time	12-hour clock, including seconds and AM/PM display
Short Time	24-hour clock, not including seconds

Create your own mask

1. In the **Input Mask** field, enter the mask syntax. This defines the length, character type, fill character, and other information. Semi-colons separating the second and third sections are automatically added to the mask.
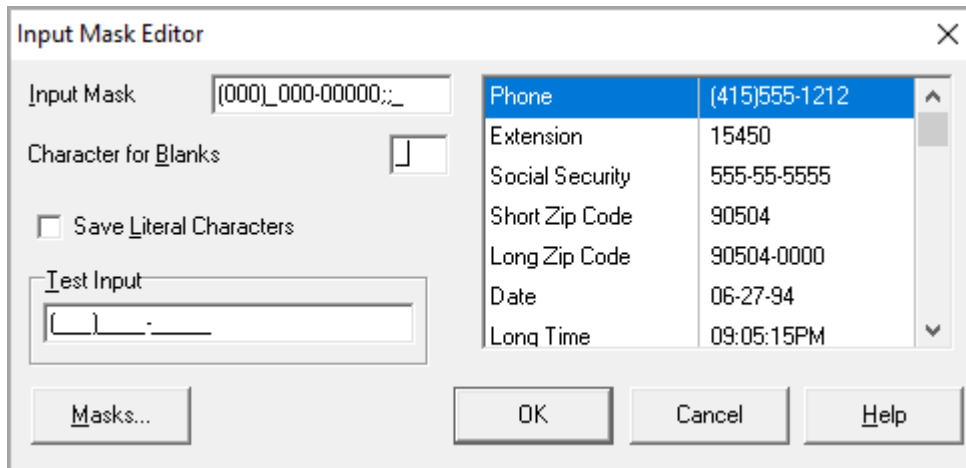
2. By default, an underscore is used as the character for blanks. To use a different character, enter that character in the **Character for Blanks** field. The input mask syntax changes accordingly.

> ⓘ By default, this is the same as the character that stands for literal spaces. The two characters (empty space and literal space) appear the same in a box in the Validate and Escalate stations. However, when a user edits the text in these stations, the cursor selects each blank character in turn, and skips over the space character.

3. Literal characters are inserted automatically, and the cursor skips over them during editing in the Validate and Escalate stations. To store the literal characters from the edit mask as part of the data, select the **Save Literal Characters** check box. The input mask syntax changes accordingly.

   A 0 in the second section of the mask syntax indicates that literal characters should not be included in the box (the box which contains characters to be edited) in the Validate and Escalate stations. Any other character indicates that they should be included.

Remove a mask

To remove a mask, clear the contents of the **Input Mask** field.

Test a mask

Use the **Test Input** box to verify that your mask works. This box displays the edit mask as it will appear on the form in the Validate and Escalate stations.

Import a mask

To import a mask (a .dem file), click **Masks**, select the file and click **Open**.

Lookup table properties

Lookup tables facilitate data entry in the Validate station by providing a selection list of possible field values. Known information, such as names, addresses, and identification numbers, can serve as predefined values that operators can select to replace unrecognized data. Values entered by an operator can be validated against the list of permissible values. Each item in the lookup table is associated with an appropriate field in a corresponding form. See Define lookup tables for information on how to create a lookup table.

Property	Description
LookupTable	Select a lookup table to associate with the field.
LookupItem	Select a lookup table item to associate with the field.
	After you have associated an item with a field, any field appearing later in the hierarchy tree may only receive the lookup table items following the one assigned.
	For example, there are three fields in one field group, and three items in a lookup table. If the first field is assigned lookup table item 2, the second field may only be assigned lookup table item 3. Because the second field is assigned lookup table item 3, then the last field will not have a lookup table item available.
	The lookup table items should appear in the same order that they appear in the ASCII file containing the data.
ShowListbox	Select the key that the operator can press to open the lookup selection list.
Automatic Fill	Select an option to complete a value from the lookup table when the operator enters sufficient characters to enable the field's unequivocal identification.

Property	Description
ForceValidation	Select this check box to validate that the field value entered by the operator exists.

Tile properties

The Tile module is an optional station usually located before the Validate station to make the Validate station more efficient. The Tile module groups per character all the characters that were recognized by the system and offers useful tools to quickly and easily determine which characters were falsely recognized. Using the Tile module, it is possible to identify which characters are correct and which are not. The Tile module displays the characters in a tile format, which allows the user to browse through each character and accept it or reject it. All characters are grouped together to allow for easy identification.

Select the **Supported** box to enable the Tile module. If the **Tile** property is enabled, the following properties appear:

Property	Description
Character filter	Click ⋯ to open a virtual keyboard from which you can select characters to be displayed in the Tile station.
Max. Char. Confidence	Enter the minimum recognition confidence level for a specified character to be displayed in the Tile station. The default is 100%, meaning that all the specified characters will be displayed.

Bad Character Level properties

The **Bad Character Level** property allows you to specify a maximum number of unrecognized characters for the field. If this value is exceeded, the whole field will appear in the Validate station as a single unrecognized character (*). It may simply be faster for the operator to visually check the data from the field and type it, rather than trying to correct a field with too many unrecognized characters (asterisks). If enabled, this property is typically set to 3 or more.

Bad Character Level	Description
For Display	Set the number of unrecognized characters that causes the whole field to appear in Validate as one unrecognized character (*).
For Process	Set the number of unrecognized characters that causes processing to stop and the whole field to appear in Validate as one unrecognized character (*).

Freedom properties

Property	Description
Learning display rule	This rule sets the condition under which to display the field in the Learning station. • **If confidence less 25%** • **If confidence less 50%** • **If confidence less 75%** • **If confidence less 100%** • **Always**: All fields are sent to the Learning station (recommended) • **Never**: No learning will take place • **When unrecognized**: Only unrecognized fields are sent to the Learning station
Confidence Threshold	Specify a confidence threshold in percent. If the confidence is below this threshold, the field will be treated as if validation had failed and will be presented in the Validate station for manual completion.

Other properties

Property	Description
Export Field	Select this check box to export this field for archiving. Some fields may be useful for the purposes of validation, but not necessary for export; this check box should be deactivated for such fields.

Property	Description
Exceptions	Specify the available exceptions for the field. Exceptions allow Validate station operators to leave invalid data in a field if they are unable to correct it. The exceptions can be resolved by a supervisor in the Escalate station. See Define exceptions for more information. There are two ways to link exceptions. Choose either option A or option B. • Option A: Expand **Exceptions** and for each exception, select **True**. • Option B: Click [...] to the right of the **Exceptions** property. In the **Exceptions** dialog box, check the check box for each exception, then click **OK**. The selected exceptions are automatically set to **True** in the Object Inspector.
BuiltIn rules	Override the default behavior of built-in rules. Click on the + button to the left of the **BuiltIn rules** property to display and change the existing rule defaults: • **FieldEmptyCheck** • **FieldLengthCheck** • **FieldValueTypeCheck** • **FieldRangeCheck** • **FieldCompletionCheck** • **LookupTableCheck** Values are **True** and **False** (default) to eliminate the check.

Property	Description
ROI delimiter	Click ⋯ to open a virtual keyboard from which you can select a character (such as - or /) that will function as a separator when there are multiple ROIs in a field. The ROI is the Region of Interest representing the area of the form in which the field appears.
Virtual engine	Select the recognition engine that will be used to recognize contents of the field. This will set the default virtual OCR engine for the regions. To override the default, enter a different virtual OCR engine under the ROI properties **Virtual engine** property. The virtual engine is the group of OCRs used to process and recognize the data in the field. See Design virtual engine for more information.
Doc Merge Auto Action	Specify from where in a multi-page document to take recurring field values (such as an invoice number or date that appears on multiple pages): • **First page** • **Last page** • **Any but first page** • **Any but last page** • **First occurrence** • **Last occurrence**
Multi Document	Select this check box to accept multiple documents that contain the same ROIs.

Associate rules with a field

You can define rules for a field.

Almost every form has rules concerning the validity of data in the form (such as minimum or maximum values, mathematical relationships between fields, and so on). These validation rules are defined during the setup process. These rules are especially important, as data entered in the fields during the data completion process must also obey these rules. The Validate station will prevent an operator from proceeding to the other fields until the contents of a field are corrected or, when the data is invalid and cannot be corrected, the operator marks an exception.

1. In the Application Explorer tree, click to select the field object.

2. Right-click the highlighted object to open the context menu and select **New Rule**.

   The rule appears in the application object tree under the field. The Object Inspector displays the properties and events.

3. Type a name and press Enter.

4. Set the rule properties. See Rule properties for more information.

Associate validation functions with a field

Validation checks allow for sophisticated, automatic checking of field contents at many different levels of inquiry. Not only can fields be related, but they may depend on one another. For example,

zip codes, city names, and state names are related. Similarly, telephone area codes and city names have a certain relationship since telephones area codes are geographically determined. Date fields may be logically limited to certain dates, with any other dates being considered invalid. For example, if, in a current census, a respondent answered that he was born in 1840, the date field could not be considered valid. Validation functions can be designed to perform these sorts of checks.

1. In the Application Explorer tree, click to select the field object.

2. Right-click the highlighted object to open the context menu and select **New VFunction**.

   The validation function appears in the application object tree under the field. The Object Inspector displays the properties and events.

3. Type a name and press Enter.

4. Set the validation function properties. See Validation function properties for more information.

Field events

You can assign events to a field in the **Events** tab of the Object Inspector. Events are actions that are triggered at a given station when the object is entered, exited, or a hotkey is pressed. .NET or Win32 DLLs are attached to these events to run special actions.

Typical events may be inserting a necessary character not included in the form data, deleting an unnecessary character included in the form data, changing a common, known incorrect character to a known correct one, and so on.

See Event functions for information on the events available for fields and how to assign events.

Chapter 7

Design table

Tables are used when the physical form includes data in the form of a table. An important feature of a table is that all the values in a column belong to the same data field. In a table, the data field is defined in the column heading. A column is equivalent to a regular field. The difference is that it usually contains more than one value (row) per field (column).

Defining a table in the Application Designer involves defining the physical location, structure and boundaries of the table in the physical form. In other words, you must draw the table on the EFI.

After a table is defined, the operator in the Validate station can jump between values in the table using the Tab key, in a logical order. This makes it easier to go through the data. The field order as it appears in the Application Designer object tree is reflected exactly in the field order in Validate station.

Create a table

Tables can only be added to a field group.

1. In the Application Explorer tree, click to select the field group object.

2. Right-click the highlighted object to open the context menu and select **New Table**.

    The table appears in the application object tree. The Object Inspector displays the properties and events.

3. Type a name and press Enter.

4. Click + to the left of the new table. A new column is added automatically when you create a table. To add more columns:

    a. Right-click on the table object and select **New Column**.

    b. Type a name and press Enter.

5. To define table rows, click on the table object to select it. In the Object Inspector, in the **Rows** field, enter the number of rows that will be processed in the table.
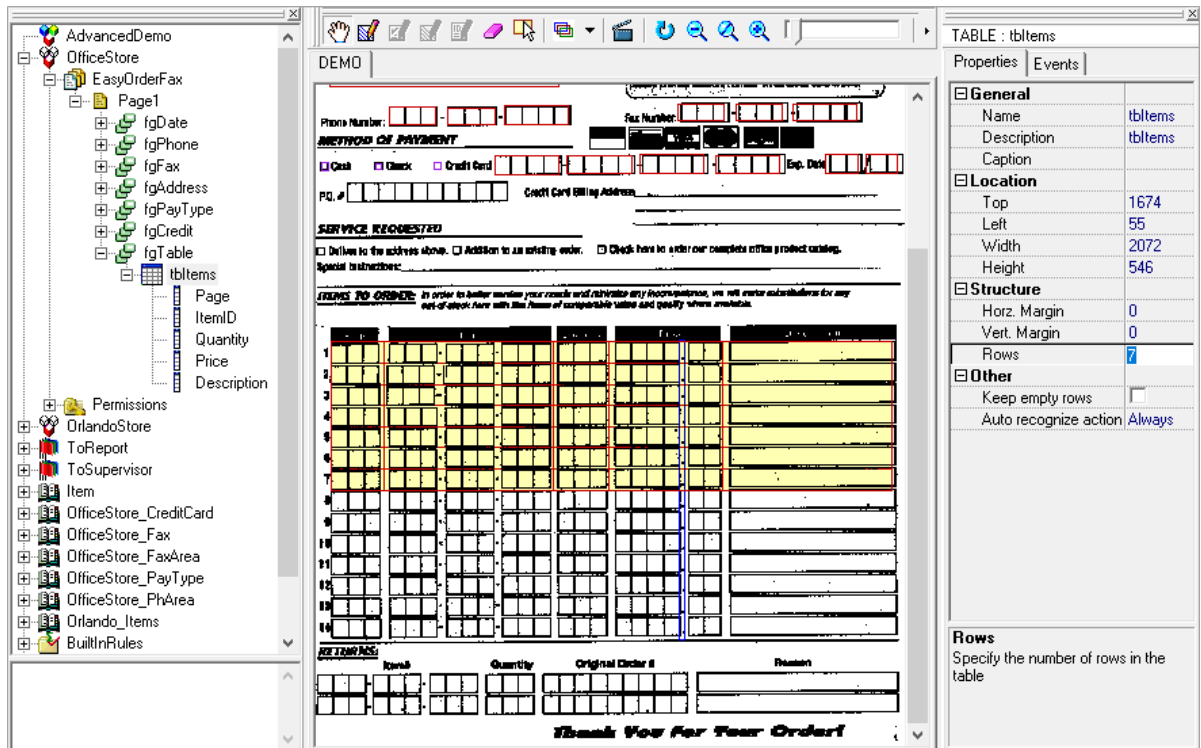
6. Draw an ROI for the table:

    a. On the Working Area toolbar, click **Draw OCR ROI** .

    b. Position the mouse pointer on the image, and then press and hold the left mouse button.

    c. Drag the mouse pointer to create a rectangle over the desired region, until you reach the size that you want.

    d. Release the mouse button.

7. Position the mouse cursor on the column separators and manually drag them to the correct location (if needed).
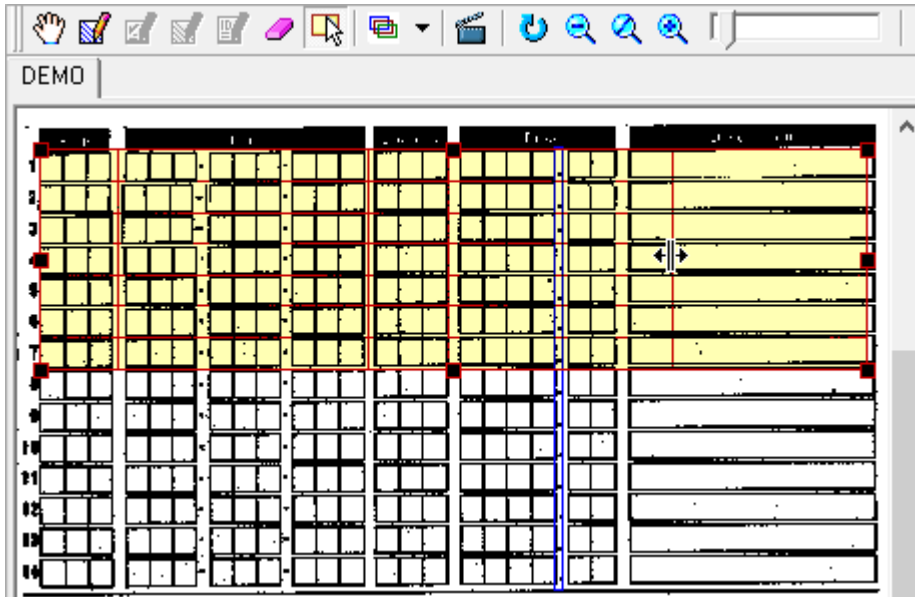
8. Set the table properties and the column properties for each column.

Table properties

General properties

Property	Description
Name	The name of the table. The name should be alphanumeric. This name appears automatically in the **Description** box in the Object Inspector window after you create a table. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the table.
Caption	A name for the table that appears on the screen throughout the application.

Location properties

The table location properties are automatically entered in the Object Inspector when you create an OCR region for the table. All values are in pixels.

For more information about creating regions for a table, see Define table regions.

Location	Description
Top	The distance from the top of the page.
Left	The distance from the left side of the page.
Width	The width of the table.

Location	Description
Height	The height of the table.

Structure properties

Property	Description
Horizontal Margin	Specify a margin between the columns of a table.
Vertical Margin	Specify a margin between the rows of a table.
Rows	Specify the number of rows expected in the table.

Create a column

A column in a table is like a field, except that it contains many data entries per form, one for each row in the column.

Columns can only be added to a table object.

1. In the Application Explorer tree, click to select the table object.
2. Right-click the highlighted object to open the context menu and select **New Column**.

   The column appears in the application object tree. The Object Inspector displays the properties and events.
3. Type a name and press Enter.
4. Set the column properties.

Column properties

General properties

Property	Description
Name	The name of the column. The name should be alphanumeric. This name appears automatically in the **Description** box in the Object Inspector window after you create a column. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the column.
Caption	A name for the column that appears on the screen throughout the application.
Label	The label that is displayed next to the column field in manual stations.
Type	The type of data permitted in the column. See Field Type settings for more information.

Property	Description
Mode	Specify the display mode for the column in the Validate station. • **Standard**: A field that can be filled in. • **Check Group**: Displayed as check boxes. More than one option can be selected. (Only for electronic forms).
Max Lines	Maximum number of permitted lines in the column.

Validation properties

Specify what sort of validation checks will be performed on the data found in the column.

Validation	Description
Double Typing	Select this check box to specify that completion results will be accepted after being confirmed at the Validate station. (That is, two operators must confirm results.)
ReadOnly	Select this check box to specify that the field is read-only and cannot be edited.
Auto Tab	Select this check box to automatically move the cursor to the next field when the Validate station operator enters the maximum number of characters in the field.
EditMask	Specify a mask to define which characters are acceptable in the column, and how the column will be displayed in the Validate and Escalate stations. See Edit Mask settings for more information.
Display Rule	Specify under what conditions a field will be displayed in the Validate and Escalate stations. Values: **Always**, **Never**, **When Empty**, **When Not Empty**, **When Unrecognized**, **Once**.
Value Required	Select this check box to specify that the field must contain a value to be considered valid.

Lookup table properties

Lookup tables facilitate data entry in the Validate station by providing a selection list of possible field values. Known information, such as names, addresses, and identification numbers, can serve as predefined values that operators can select to replace unrecognized data. Values entered by an operator can be validated against the list of permissible values. Each item in the lookup table is associated with an appropriate field in a corresponding form. See Define lookup tables for information on how to create a lookup table.

Property	Description
LookupTable	Select a lookup table to associate with the field.
LookupItem	Select a lookup table item to associate with the field.
	After you have associated an item with a field, any field appearing later in the hierarchy tree may only receive the lookup table items following the one assigned.
	For example, there are three fields in one field group, and three items in a lookup table. If the first field is assigned lookup table item 2, the second field may only be assigned lookup table item 3. Because the second field is assigned lookup table item 3, then the last field will not have a lookup table item available.
	The lookup table items should appear in the same order that they appear in the ASCII file containing the data.
ShowListbox	Select the key that the operator can press to open the lookup selection list.
Automatic Fill	Select an option to complete a value from the lookup table when the operator enters sufficient characters to enable the field's unequivocal identification.
ForceValidation	Select this check box to validate that the field value entered by the operator exists.

OCR properties

Property	Sub-property	Description
MaxChars		Maximum number of characters expected to be found in the column field.

Property	Sub-property	Description
MinChars		Minimum number of characters expected to be found in the column field.
MaxWords		Maximum number of words expected to be found in the column field.
MinWords		Minimum number of words expected to be found in the column field.
ChangeCase		Choose from the following case-sensitivity options for processing column field contents: • **Do not change** • **Change to upper case**: Change all contents to upper case • **Change to lower case**: Change all contents to lower case
Padding Info		Select when you want the column field to always have a certain character or number length (for example, cents would always be 2 characters). Padding characters can only be inserted at the beginning or end of a string, such as lead or decimal zeros.
	Character	Chose the character to be used for padding empty spaces, such as 0 for decimal values. Click ⋯ to the right of the property to open a virtual keyboard from which you can select characters. Select the character to be excluded.
	Direction	Specify character alignment, for example, justified, left or right.
	Length	Specify the number of characters used for padding.
Insert Chars		Specify characters to be inserted into the column field during processing. For example, a zip code or area code may be automatically added to a field.
	Position	Select the start position where the characters are to be inserted during processing.
	String	Specify which character will be inserted during processing.
Replace Chars		Define these settings to replace a specified set of characters during processing. For example, changing a common, known incorrect character to a known correct one, such as changing x to %.
	Position	Select the start position where the characters are to be replaced during processing.
	String	Specify which characters will be replaced during processing.

Property	Sub-property	Description
ExcludeChars		Select characters that should be excluded from the column field during processing.
		Click ⋯ to the right of the property open a virtual keyboard from which you can select characters. Select the character to be excluded.
		For example, you may want to eliminate from the field characters such as # that are not relevant data.
Virtual Engine		Select the default OCR virtual engine to be used in the Recognize station for character recognition for the column field.

Tile properties

The Tile module is an optional station usually located before the Validate station to make the Validate station more efficient. The Tile module groups per character all the characters that were recognized by the system and offers useful tools to quickly and easily determine which characters were falsely recognized. Using the Tile module, it is possible to identify which characters are correct and which are not. The Tile module displays the characters in a tile format, which allows the user to browse through each character and accept it or reject it. All characters are grouped together to allow for easy identification.

Select the **Supported** box to enable the Tile module. If the **Tile** property is enabled, the following properties appear:

Property	Description
Character filter	Click ⋯ to open a virtual keyboard from which you can select characters to be displayed in the Tile station.
Max. Char. Confidence	Enter the minimum recognition confidence level for a specified character to be displayed in the Tile station. The default is 100%, meaning that all the specified characters will be displayed.

Bad Character Level properties

The **Bad Character Level** property allows you to specify a maximum number of unrecognized characters for the field. If this value is exceeded, the whole field will appear in the Validate station as a single unrecognized character (*). It may simply be faster for the operator to visually check the data from the field and type it, rather than trying to correct a field with too many unrecognized characters (asterisks). If enabled, this property is typically set to 3 or more.
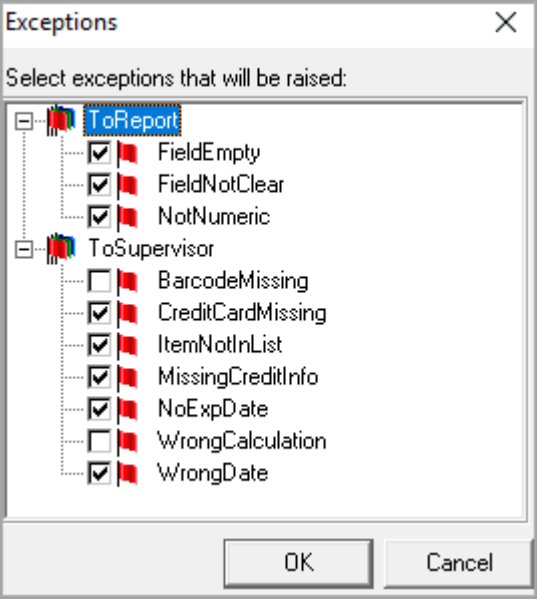
Bad Character Level	Description
For Display	Set the number of unrecognized characters that causes the whole field to appear in Validate as one unrecognized character (*).
For Process	Set the number of unrecognized characters that causes processing to stop and the whole field to appear in Validate as one unrecognized character (*).

Freedom properties

Property	Description
Learning display rule	This rule sets the condition under which to display the field in the Learning station. • **If confidence less 25%** • **If confidence less 50%** • **If confidence less 75%** • **If confidence less 100%** • **Always**: All fields are sent to the Learning station (recommended) • **Never**: No learning will take place • **When unrecognized**: Only unrecognized fields are sent to the Learning station
Confidence Threshold	Specify a confidence threshold in percent. If the confidence is below this threshold, the field will be treated as if validation had failed and will be presented in the Validate station for manual completion.

Other properties

Property	Description
Export Field	Select this check box to export this field for archiving. Some fields may be useful for the purposes of validation, but not necessary for export; this check box should be deactivated for such fields.

Property	Description
Exceptions	Specify the available exceptions for the field. Exceptions allow Validate station operators to leave invalid data in a field if they are unable to correct it. The exceptions can be resolved by a supervisor in the Escalate station. See Define exceptions for more information. There are two ways to link exceptions. Choose either option A or option B. • Option A: Click + to the left of the **Exceptions** property. In the dropdown list for each exception, select **True**. • Option B: Click ⋯ to the right of the **Exceptions** property. In the **Exceptions** dialog box, select the check box for each exception, then click **OK**. The selected exceptions are automatically set to **True** in the Object Inspector.
BuiltIn rules	Override the default behavior of built-in rules. Click + the left of the **BuiltIn rules** property to display and change the existing rule defaults: • FieldEmptyCheck • FieldLengthCheck • FieldValueTypeCheck • FieldRangeCheck • FieldCompletionCheck • LookupTableCheck Values are **True** and **False** (default) to eliminate the check.

Property	Description
ROI delimiter	Click the selection button ⋯ to open a virtual keyboard from which you can select a character (such as - or /) that will function as a separator when there are multiple ROIs in a field. The ROI is the Region of Interest representing the area of the form in which the field appears.
Virtual engine	Select the recognition engine that will be used to recognize contents of the field. This will set the default virtual OCR engine for the regions. To override the default, enter a different virtual OCR engine under the ROI properties **Virtual engine** property. The virtual engine is the group of OCRs used to process and recognize the data in the field. See Design virtual engine for more information.
Doc Merge Auto Action	Specify from where in a multi-page document to take recurring field values (such as an invoice number or date that appears on multiple pages): • **First page** • **Last page** • **Any but first page** • **Any but last page** • **First occurrence** • **Last occurrence**
Multi Document	Select this check box to accept multiple documents that contain the same ROIs.

Column events

You can assign events to a column. Events are actions that are triggered at a given station when the object is entered, exited, or a hotkey is pressed. .NET or Win32 DLLs are attached to these events to run special actions.

Typical events may be inserting a necessary character not included in the form data, deleting an unnecessary character included in the form data, changing a common, known incorrect character to a known correct one, and so on.

See Event functions for information on the events available for columns and how to assign events.

Design EFI

A physical form is a structured document that includes one or more pages to be filled with data. Physically, a form may have many variations - handwriting and printing anomalies, different sizes, slightly altered logos, differences in the exact location or size of the fields on the page, and so on. The same form, printed by different printers, may have a different ink color, contain minor printing differences, or have information or fields placed in slightly different locations. For example, a bar code on one form may appear fractionally higher than on another form. Several editions of the same form may not be the same size. However, despite minor physical differences, all the variations still represent the same form.

eFlow allows you to associate several physical appearances to a single logical form. Each physically distinct appearance of an empty form is called an Empty Form Image, or EFI. The EFI is a template of the structured document, which the system uses to extract the data. The eFlow term EFI is a logical definition and simply means images of the empty forms. An image file should be made for every physically distinct empty form.

A form may also contain one or more pages. In many cases, a form has several variations, so you can assign several EFIs to the same form or page. In fact, for every logical page in the form, you can define one or more EFIs. Each single EFI is an image of an individual physical page within the form.

EFI regions

Each form must go through several stages of processing. First, the type of form must be established, then the data in the form must be examined and recognized.

To accomplish these tasks, you must identify parts of the form in different ways. Some parts of the form help to identify which form it is. Other parts of the form are examined for the data that has been filled in.

These parts of the forms are called Regions of Interest (ROIs). Regions of Interest are the regions of the scanned form that will undergo various types of data processing. The basic procedure for defining Regions of Interest and their parameters is as follows:

1. Delineate each Region of Interest in the EFI.

2. For each Region of Interest, define parameters according to what purpose it will serve.

Just as EFIs are linked to specific pages, Regions of Interest are linked to either pages, fields, or columns, depending on what purpose they serve. FormID and removal regions, which are used for form identification, are linked to pages. OCR and OMR regions (the regions where data recognition is performed) are linked to fields or tables. See Define page level regions and Define field level regions for more information.

Create an EFI

You can create an EFI from a pre-scanned image file, or by scanning a form.

Create an EFI from an image file

If you already have an image file of the empty form, you can open the file and save it as an EFI file. EFIs are saved with the file extension .efi.

1. In the **Tools** menu, select **Get New EFI from File**.

2. In the **Open** dialog box, click on the file name to select the image file.
   The image appears in the preview panel on the right of the dialog box.
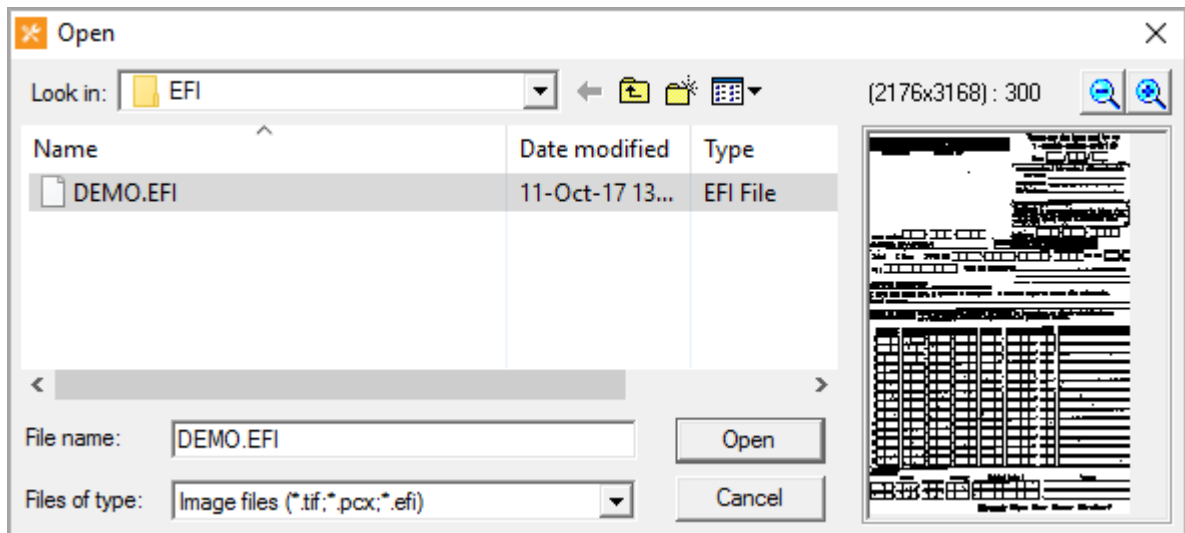
3. Click **Open**.

4. In the **Save As** dialog box, type a name and click **Save**.

Scan an EFI from a paper form

eFlow provides scanning support in the Scan module and the Application Designer. During application setup using the Application Designer, you can directly scan an empty form; the system will learn this form and create a new EFI for it. You can also directly scan test batches to test EFI form recognition. See Test an EFI for more information.

Save an EFI

EFIs that you create are automatically saved when you save the entire application. You can save the EFI independently, but it is entirely optional.

1. In the Working Area, click to select the EFI.

2. Right-click the EFI to open the context menu and click **Save EFI**.

Link an EFI to a page

1. In the Application Explorer tree, click to select the page object.

2. In the Object Inspector, click ⋯ to the right of the **EFI** property.

   The **EFIs Linking** dialog box opens. In the application tree, under each page is a list of EFIs with check boxes appear.

3. To view an EFI, click on the name of the EFI (not the check box).

   The selected EFI is displayed in the preview panel on the right.
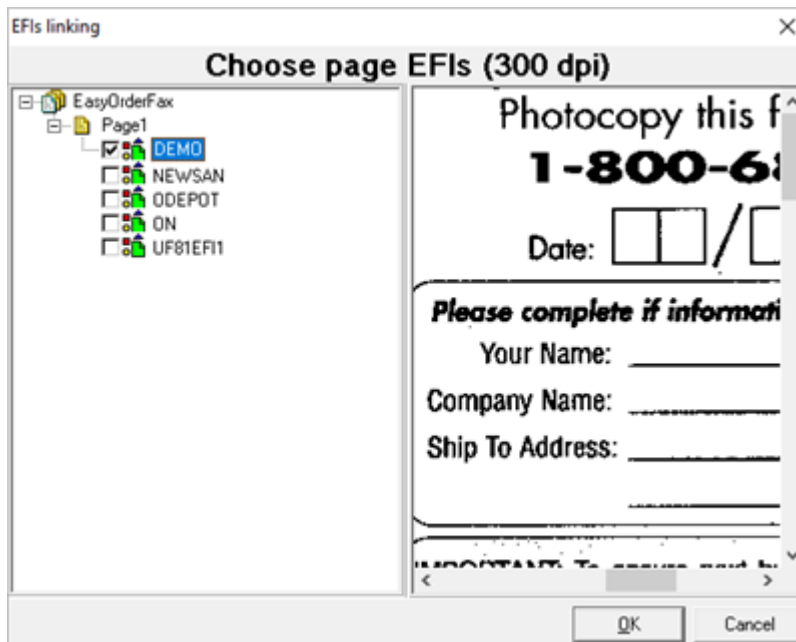
4. Select the check box next to an EFI to link it to the page. You can link more than one EFI to a page.

5. Click **OK**.

To remove the link between an EFI and the page, clear the check box next to the EFI.

EFI properties

To set EFI properties, click on the EFI in the Working Area to select it. The Object Inspector displays the properties and events.

Tolerances

Tolerance is the level of difference allowed between the EFI and the TIF forms on which removal is performed. If the difference exceeds specific tolerances, the form may not match the EFI and will be declared unrecognized.

1. Click ... to the right of the **Tolerances** property.

    The **Set EFI Tolerance Parameters** dialog box opens:
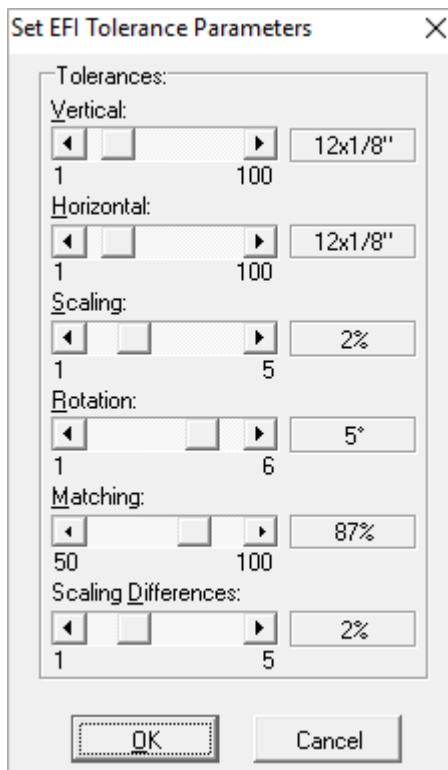
2. To set the tolerances, drag the slide bar, or click the arrows to change the current tolerance settings. The values displayed on the right change accordingly.

3. Click **OK**.

Parameter	Range	Unit	Description
Vertical	1-100	1/8"	Vertical shift between the TIF image and the EFI.
Horizontal	1-100	1/8"	Horizontal shift between the TIF image and the EFI.
Scaling	1-5	1%	Amount by which the TIF size may differ from the EFI size.
Rotation	1-6	1 °	Angle that the TIF may be rotated relative to the EFI.

Parameter	Range	Unit	Description
Matching	50-100	1%	Difference between the EFI and the TIF.
			The default value is 87%, which is strict, but normal results are usually better (98-99%). The default matching value is suitable for most uses. Only change this value in special cases. For example, if you know (or can see from the test results) that the forms do not match the EFI very well, you may wish to set the matching property tolerance lower.
Scaling Difference	1-5	1%	Difference between the vertical scaling and the horizontal scaling.

Processing properties

The **Processing** properties allow you to define options that aid form recognition.

Property	Description
Registration	Select this check box when forms contain a small amount of constant data, such as lines and boxes.
NotOriginal	Select this check box when forms are of poor-quality, such as photocopied forms. This option is also recommended when the scanner is suspected of producing poor-quality images (for example, fax).
BadBeginning	The recognition engine compares the filled-in form with the empty form starting at the top left corner of the form. When the engine concludes that there is no match between the filled-in form and the EFI, it stops the comparison.
	Select this check box if you expect that the top left corner may cause problems during the comparison (for example, it is dirty, there is extraneous information there, and so on). In this case, the engine stops the comparison immediately.
BadMargins	Select this check box to instruct the recognition engine that the forms were truncated during scanning (or the paper itself is truncated).

Enhancement properties

Property	Description
Before FormID	Click ⋯ to define the enhancement filters that are applied to each page before the page is identified. The FormID algorithm allows the Recognize module to recognize the type of form and process it accordingly. One or more regions in the EFI are defined as form recognition regions, in the case where there are several types of forms that differ only by these regions. If at least one FormID region does not comply with predefined conditions, the form is not matched to the EFI. See Define page level regions for more information.
	In certain cases, the quality of the form image may make recognition difficult. Setting enhancement filters can help to improve form recognition. See The Enhancement Builder for more information.

Other properties

Property	Description
Recognition Engine	Select an algorithm to determine how the EFI is removed from the REG (Registered Image, or pre-processed TIF) to produce the DIF (Difference) file. • **FormOut** performs form identification and standard form removal (that is, it removes the EFI from the REG to produce the DIF). • **FeaturesID** uses an alternative algorithm and almost always recognizes the forms but does not perform form removal. It is less sensitive than FormOut! to small changes in the form. This algorithm is very rarely used. • **DropOut** uses yet another algorithm to identify forms that have had EFI details removed during scanning (for example, if a special ink and color bulb is used to remove the EFI during the scanning process). You must therefore provide additional information for processing. ⓘ If the form is not a DropOut form, use FormOut! Only if it fails should you try FeaturesID.
Default Algorithm	The default algorithm determines the type of removal that is performed on the entire filled form. You can also specify the removal algorithm for a specific ROI, if you are using the FormOut! recognition algorithm. • **Keep Everything**: Retain both constant and variable data in the DIF file. This option is used if certain data in the EFI, such as decimal points or slashes or dashes, are necessary to the field. • **Keep Nothing**: Retain neither constant nor variable data in the DIF file (it will be blank). This option is used in conjunction with removal ROIs to include only selected areas of the form in the DIF file. • **Remove**: Retain only variable data in the DIF file (that is, remove the EFI from the REG to produce the DIF). This is the default. • **Safe Remove**: Retain variable data and constant data that overlap the variable data in the DIF file. The amount of constant data included in the DIF file is defined by the Integrity Radius parameter. This algorithm is used in cases such as handwritten signatures, where important data may appear outside the ROI.
Line Connection	This parameter is only available if you select the **Remove** or **Safe Remove** recognition algorithm. When variable data overlap constant data, lines in the variable data may be cut. Line connection reconnects the broken lines. The **Line Connection** parameter specifies whether the line connection algorithm should be applied to gaps created by the removal of overlapping data during the removal process. Line connection is therefore relevant only to areas where the **Remove** or **Safe Remove** algorithm is applied. Line connection is valuable if OCR or ICR is to be applied to the form because connected lines are much easier for OCR engines to read than broken lines. • **On**: Reconnects lines broken during form removal. • **Double Horizontal**: Improves the line connection process when there are two horizontal lines very close together. • **Double Vertical**: Improves the line connection process when there are two vertical lines very close together.

Property	Description
Integrity Radius	This parameter is only available if you select the **Remove** or **Safe Remove** recognition algorithm.
	The integrity radius specifies the radius of the information that should be preserved around a point where variable and constant data overlap. The integrity radius is therefore relevant only when the **Safe Remove** or the **Remove** algorithm is applied.
	Filled-in information often overlaps the constant data of the form. This could lead to the loss of sensitive information (such as part of a signature) during removal of the constant data. To prevent the loss of such information, apply the **Safe Remove** algorithm and **Integrity Radius** to such areas, to retain all overlapping constant and variable data within the integrity radius.
	For example, when someone fills in a form, a letter might overlap the box that is filled in. The constant data is the lines of the box, and the variable data is the filled-in information.
	Specify the integrity radius from the **Integrity Radius** list box buttons in the dialog box. The options retain all the data within the integrity radius from the intersection of constant and variable data. • **Low**: 0.07 cm • **Medium**: 0.14 cm • **High**: 0.21 cm
Try Rotations	This FormOut! feature is used when recognition has initially failed. It allows you to set the angle of rotation (90°, 180° and 270°) of the form as the system attempts to recognize the form again.
	This parameter helps to identify the EFI that matches the form. If the form has been rotated during scanning, eFlow detects misalignment and rotates the form before attempting to match it with an EFI.
	For example, if no presorting is done and the forms may be input upside down, select both the 0° and 180° options. ⓘ This option can slow down form identification, so use it sparingly.
Caption	

EFI toolbar

The EFI Toolbar is displayed above the Working Area in the Application Designer. It provides tools for setting Regions of Interest to aid form identification and field recognition. Position the mouse cursor over a button to display a tool tip that explains its purpose.

Different buttons are enabled for page objects and for field or table objects. This is due to the different recognition needs at the page, field, and table level.

Icon	Name	Description	Enabled for	Refer to
	Pan image	Move the EFI around in the Working Area (when the Working Area does not display the entire EFI).	Page, group, field	
	Draw OCR ROI	Specify which areas contain field data for which OCR recognition will be performed.	Field	Define an OCR region
	Draw OMR ROI	Specify which field areas contain check marks for improved recognition.	Field	Define an OMR region
	Draw Removal ROI	Specify areas of the form that will be removed in the Recognize station.	Page	Define a removal region
	Draw FormID ROI	Specify areas of consistent and constant data that can reliably be used to identify the form during the recognition phase.	Page	Define a FormID region
	Erase regions	Delete a region from the EFI.	Page, group, field	Delete an ROI
	Selection mode	Select a region in the EFI.	Page, group, field	Select an ROI

Icon	Name	Description	Enabled for	Refer to
	Region type filter	Specify which regions to display on the EFI. Click the arrow and then select which ROIs you want to display, or click the button repeatedly until the desired region is selected: • Red: OCR/OMR regions • Blue: Removal regions • Green: FormID regions • All colors: All regions • White: No regions	Page, group, field	EFI regions
	FormOut Test	FormOut is a recognition algorithm. Click this button to test the FormOut effectiveness for the scanned form.	Page, group, field	Test an EFI
	Rotate	Rotate the EFI image. Each click rotates the image clockwise by 90°. ⓘ This rotation is not retained in the EFI file.	Page, group, field	
	Zoom out	Shrink the EFI display by 10%.	Page, group, field	
	Actual size	Return the EFI display to 50%.	Page, group, field	
	Zoom in	Enlarge the EFI display by 10%.	Page, group, field	
	Zoom bar	Specify the zoom percentage at which the EFI is displayed. Use the slider or click in the bar to shrink or enlarge the image in increments.	Page, group, field	

Define page level regions

Page level Regions of Interest enable eFlow to identify the form. eFlow supports separate algorithms for different kinds of form identification.

• FormID algorithm

It is recommended that you define FormID for one or more regions in the EFI when there are several types of forms that differ only by these regions. When a batch of forms is received, eFlow locates the corresponding EFI by performing FormOut! tests, that is, by checking the FormID regions.

The FormID tests are quite sensitive. If at least one FormID region does not comply with the predefined conditions, the form is not matched to the EFI.

> ℹ️ Be careful to select regions that contain the same information for all of the forms that you wish to identify as being of the same type.

- Removal algorithm

  A default removal algorithm is defined for the entire form. In addition, a different removal algorithm can be defined for each Region of Interest that is defined for removal.

Define a FormID region

FormID is defined for the EFI at the page level.

1. In the Application Explorer tree, click to select the page object.
2. On the **EFI** toolbar, click **FormID** 📝.
3. Point at the image and click and hold the left mouse button.

   The pointer becomes a cross.
4. Drag the cross to create a rectangle over the desired region, until you reach the size that you want.
5. Release the mouse button.

A green rectangle appears over the region. The ROI properties are displayed on the right in the Object Inspector.

FormID region properties

Set the following properties through the Object Inspector.

Property	Sub-property	Description
Top		Select the starting point setting for the ROI. Defined automatically when you create or resize the ROI manually.
Left		Select the distance from the left for the ROI. Defined automatically when you create or resize the ROI manually.
Width		Specify the ROI width. Defined automatically when you create or resize the ROI manually.
Height		Specify the ROI height. Defined automatically when you create or resize the ROI manually.
Slice X	Columns	Split the field vertically into two or more ROIs. Select how many columns to create with the OCR's ROI.
	Margin	Define the distance between each column.
	Slice	Split the OCR's ROI vertically into a few columns, as defined by the value in the **Columns** property.
Slice Y	Rows	Split the field horizontally into two or more ROIs. Select how many rows to create with the OCR's ROI.
	Margin	Define the distance between each row.

Property	Sub-property	Description
	Slice	Split the OCR's ROI horizontally a few rows, as defined by the value in the **Rows** property.
Other	FormID Algorithm	Select a form recognition algorithm. • **Match EFI**: The contents of the Region of Interest must be the same as in the EFI • **Any data**: The Region of Interest must contain data • **Empty:** The Region of Interest must be empty ⓘ The FormID algorithm you select must match the identification region you choose.

Define a removal region

The Application Designer allows you to define a default removal algorithm. In addition, you can define a different removal algorithm for each removal Region of Interest.

1. In the Application Explorer tree, click to select the page object.
2. On the EFI toolbar, click **Draw Removal ROI** 📝.
3. Point at the image and click and hold the left mouse button.

   The pointer becomes a cross.
4. Drag the cross to create a rectangle over the desired region, until you reach the size that you want.
5. Release the mouse button.

A blue rectangle appears over the region. The ROI properties are displayed on the right in the Object Inspector.

Removal region properties

Set the following properties through the Object Inspector.

Property	Sub-property	Description
Top		Select the starting point setting for the ROI. Defined automatically when you create or resize the ROI manually.
Left		Select the distance from the left for the ROI. Defined automatically when you create or resize the ROI manually.
Width		Specify the ROI width. Defined automatically when you create or resize the ROI manually.
Height		Specify the ROI height. Defined automatically when you create or resize the ROI manually.
Slice X	Columns	Split the field vertically into two or more ROIs. Select how many columns to create with the OCR's ROI.

Property	Sub-property	Description
	Margin	Define the distance between each column.
	Slice	Split the OCR's ROI vertically a few columns, as defined by the value in the Columns property.
Slice Y	Rows	Split the field horizontally into two or more ROIs. Select how many rows to create with the OCR's ROI.
	Margin	Define the distance between each row.
	Slice	Split the OCR's ROI horizontally a few rows, as defined by the value in the Rows property.
Removal Algorithm		Select an option for form removal. • **Keep Everything**: Retain both constant and variable data. • **Keep Nothing**: Exclude any data found in the TIF image and block out form areas that should not have been written in. • **Remove**: Retain only variable data. This is the default. • **Safe Remove**: Retain the variable data and some of the constant data that overlaps the variable data. This is used when capturing signatures and other sensitive data. The amount of constant data included in the DIF file is defined by the integrity radius. • **Thin Remove**: Remove constant data with extra care within a region whose constant data overlaps variable data.
Ignore		Ignore mismatches if the region may contain constant data that differs greatly from the constant data in the EFI. The contents of such a region will not be taken into consideration during form recognition. ⓘ Use this option with care, as too many ignored regions may significantly reduce the ability of eFlow to recognize the form.

Define field level regions

Field level Regions of Interest enable eFlow to identify data that must be recognized.

• OCR regions

OCR regions are field locations where data of different types is to be recognized. For example, your form may contain fields that have dates, numbers, names, mixed text and numbers, dollar amounts, and so on. By assigning an OCR region to a field or table, you define what sort of information you expect to encounter in these fields.

OCR regions can be defined in terms of the type of text and numbers that will consistently be found in them. After you have defined a Region of Interest, you can define the OCR parameters to be applied to it using the Object Inspector.

After you have created and defined OCR regions, you allocate OCR recognition engines to each OCR region. These OCR engines are defined using the OCR Engines Designer. OCR engines can be set with sophisticated parameter control, and the OCR Engines Designer provides user-friendly tools for fine-tuning and for generating and checking recognition results.

During the runtime stage, OCR engines are used to recognize field data using individual OCR engines and virtual engines. See Design OCR engine and Design virtual engine for more information.

- OMR regions

  OMR regions allow you to specify that a field area contains check marks for improved recognition.

Define an OCR region

1. In the Application Explorer tree, click to select the field or table object.

2. On the EFI toolbar, click **Draw OCR ROI** .

3. Point at the image and click and hold the left mouse button.

   The pointer becomes a cross.

4. Drag the cross to create a rectangle over the desired region, until you reach the size that you want.

5. Release the mouse button.

A red rectangle appears over the region. The ROI properties are displayed on the right in the Object Inspector.

OCR region properties

Set the following properties through the Object Inspector.

Category	Property	Sub-property	Description
Location	Top		Select the starting point setting for the ROI. Defined automatically when you create or resize the ROI manually.
	Left		Select the distance from the left for the ROI. Defined automatically when you create or resize the ROI manually.
	Width		Specify the ROI width. Defined automatically when you create or resize the ROI manually.
	Height		Specify the ROI height. Defined automatically when you create or resize the ROI manually.
OCR	Max. Chars		Maximum number of characters expected to be found in the region.
	Min. Chars		Minimum number of characters expected to be found in the region.
	Max. Words		Maximum number of words expected to be found in the region.

Category	Property	Sub-property	Description
	Min. Words		Minimum number of words expected to be found in the region.
	Max. Lines		Maximum number of lines expected to be found in the region.
	Min. Lines		Minimum number of lines expected to be found in the region.
	Change Case		Choose from the following options for processing region contents: • **Do not change** • **Change to upper case**: Change all contents to upper case. • **Change to lower case**: Change all contents to lower case.
	Force Alphanumeric		Allows you to direct the OCR engines to get only numeric or only alpha characters as results. Choose from the following options for processing region contents: • Do not force • Force to numeric • Force to alpha
	Padding Info		Define these settings when you want the region to always have a certain character/number length. For example, cents would always be 2 characters. Padding characters can only be inserted at the beginning or end of a string, such as lead or decimal zeros.
		Char	Chose the character to be used for padding empty spaces, such as 0 for decimal values.
		Direction	Specify character alignment, choose from the following list: • justified to left • justified to right For example, cents would usually be justified right.
		Length	Specify the number of characters used for padding. For example, cents would always be 2 characters.
	Insert Chars		Specify a character to be inserted into the region during processing. For example, a zip code or area code may be automatically added to a field.
		String	Specify which character will be inserted during processing.
		Position	Select the start position where the characters are to be inserted during processing.

Category	Property	Sub-property	Description
	Replace Chars		Define these settings to replace a specified set of characters during processing. For example, changing a common, known incorrect character to a known correct one, such as changing x to %.
		Position	Select the start position where the characters are to be replaced during processing.
		String	Specify which characters will be replaced during processing.
	Exclude Chars		Click ... to select characters that should be excluded from the region during processing. For example, you may want to eliminate from the field characters such as # that are not relevant data.
	Virtual Engine		The virtual OCR engine assigned in the **Virtual Engine** property under **Field Properties** is allocated as the default, and therefore this field is normally left empty (that is, the option **None** is selected). Entering a different virtual OCR engine in this field will override the default virtual OCR engine as allocated under **Field properties**.
Slice X	Columns		Split the field vertically into two or more ROIs. Select how many columns to create with the OCR's ROI.
	Margin		Define the distance between each column.
	Slice		Split the OCR's ROI vertically into a number of columns, as defined by the value in the **Columns** property.
Slice Y	Rows		Split the field horizontally into two or more ROIs. Select how many rows to create with the OCR's ROI.
	Margin		Define the distance between each row.
	Slice		Split the OCR's ROI horizontally into a number of rows, as defined by the value in the **Rows** property.
Other	Associated Field		ROIs are linked to fields. Use this property to change the field to which an ROI is linked.
	Order in Field		For a multi-region field. Use when a field is divided into different regions. Set or change the order in which the OCR virtual engines read ROIs in the field. The order in which regions are linked to the field determines the order in which they are read.

Define an OMR region

1. In the Application Explorer tree, click to select the field or table object.
2. On the EFI toolbar, click **Draw OMR ROI** .
3. Point at the image and click and hold the left mouse button.
   The pointer becomes a cross.

4. Drag the cross to create a rectangle over the desired region, until you reach the size that you want.

5. Release the mouse button.

A red rectangle appears over the region. The ROI properties are displayed on the right in the Object Inspector.

OMR region properties

Set the following properties through the Object Inspector.

Category	Property	Description
Location	Top	Select the starting point setting for the ROI. Defined automatically when you create or resize the ROI manually.
	Left	Select the distance from the left for the ROI. Defined automatically when you create or resize the ROI manually.
	Width	Specify the ROI width. Defined automatically when you create or resize the ROI manually.
	Height	Specify the ROI height. Defined automatically when you create or resize the ROI manually.
OCR	Virtual Engine	The virtual OCR engine assigned in the **Virtual Engine** property under **Field** Properties is allocated as the default, and therefore this field is normally left empty (that is, the option None is selected). Entering a different virtual OCR engine in this field will override the default virtual OCR engine as allocated under **Field properties**.
OMR	Caption	For electronic forms. Specify a caption for the check box or radio button.
	Mark Size	If there is a mark, select the size: **msSmall**, **msMedium** or **msLarge**. If there is more than one, select **msMultiple**.
Slice X	Columns	Split the field vertically into two or more ROIs. Select how many columns to create with the OMR's ROI.
	Margin	Define the distance between each column.
	Slice	Split the OMR's ROI vertically into a number of columns, as defined by the value in the **Columns** property.
Slice Y	Rows	Split the field horizontally into two or more ROIs. Select how many rows to create with the OMR's ROI.
	Margin	Define the distance between each row.
	Slice	Split the OMR's ROI horizontally into a number of rows, as defined by the value in the **Rows** property.
Other	Associated Field	ROIs are linked to fields. Use this property to change the field to which an ROI is linked.
	Order in Field	For a multi-region field. Use when a field is divided into different regions. Set or change the order in which the OCR virtual engines read ROIs in the field. The order in which regions are linked to the field determines the order in which they are read.

Define table regions

Create an ROI for a table

To create a region for a table, first create an ROI for the entire table.

To create and define an OCR region for a table:

1. In the Application Explorer tree, click to select the table object.
2. On the EFI toolbar, click **Draw OCR ROI** .
3. Point at the image and click and hold the left mouse button.

   The pointer becomes a cross.
4. Drag the cross to create a rectangle over the desired region, until you reach the size that you want.
5. Release the mouse button.

   A red rectangle appears over the region. The ROI properties are displayed on the right in the Object Inspector.
6. By default, each table has one row. If there is more than one row in the table, specify the number of rows using the **Row** property in the Object Inspector.
7. By default, the table has one column. If there is more than one column in the table, you must create a new column object for each additional column on the EFI. Right-click the highlighted table to open the context menu and select **New Column**.

   For example, in the following image, several new columns and rows have been created for the table displayed in the Working Area:
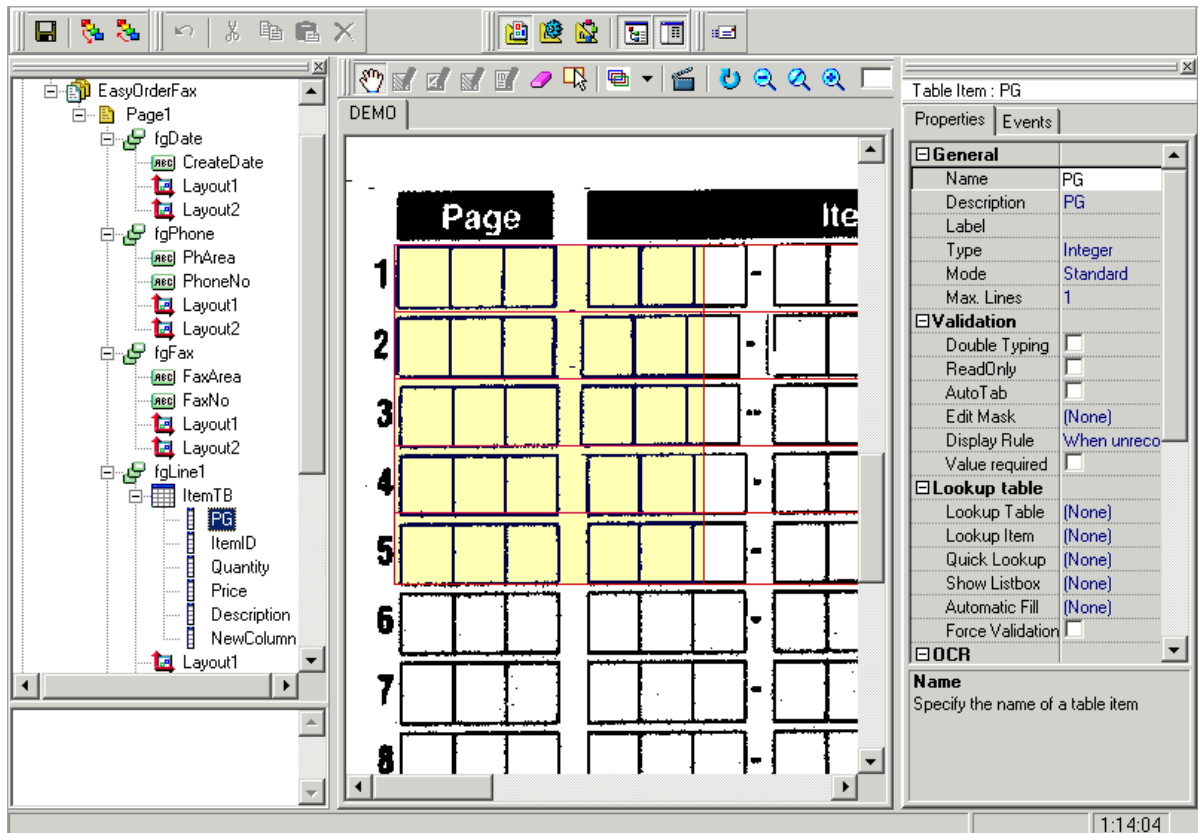
The ROI properties are displayed on the right in the Object Inspector.

Resize a column

You can change the width of a column within the table ROI.

1. Select the ROI.
   The pointer becomes a hand.

2. On the toolbar, click **Selection Mode** .

3. Click the column edge and hold the left mouse button.
   The pointer becomes a width changer pointer.

4. Drag the pointer to the new location on the screen.

5. Release the mouse button.

Move resize and delete regions

Before performing any operation on an ROI, you must first select it.

Select an ROI

1. Click **Selection Mode** .
    The pointer becomes a cross.
2. Click on the ROI in the image.

To deselect an ROI, click anywhere outside the ROI.

Move an ROI

1. Select the ROI.
2. Press and hold the left mouse button.
    The pointer becomes a hand.
3. Drag the pointer to the new location on the screen.
4. Release the mouse button.

Resize an ROI

1. Select the ROI.
2. Place the pointer on one of the eight handles.
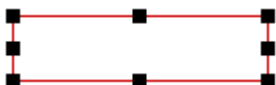
    The pointer becomes a double-headed arrow.
3. Press and hold the left mouse button.
4. Drag the pointer in the desired direction until you get the desired size.
5. Release the mouse button.

Delete an ROI

1. Select the ROI.
2. Click **Erase region** .
3. Place the pointer inside the selected region and click.

ⓘ Deleting an ROI does not delete the column objects in a table. They still exist in the Application Explorer tree.

Copy rename and delete EFIs

Copy an EFI

1. On the **Tools** menu, select **Copy EFI**.

2. In the **Open** dialog box, select the EFI file and click **Open**.

3. In the **Save As** dialog box, type a name and click **Save**. This file name must be unique.

Rename an EFI

1. In the Application Explorer tree, click on a linked page to display the EFI.

2. Right-click on the EFI in the Working Area and select **Rename EFI**.

3. In the **Save As** dialog box, type a new file name, which must be unique, then click **Save**.

Delete an EFI

1. In the Application Explorer tree, click on a linked page to display the EFI.

2. Right-click on the EFI in the Working Area and select **Delete EFI**.

Test an EFI

It is important to test EFIs to ensure that eFlow properly recognizes which EFI a form belongs to, and correctly removes the constant data (the form itself), leaving only the variable data (the filled-in part). This test process is FormOut! It is best performed on a sample of about 20 filled-in forms called "test forms".

The test returns the exact percentage of matching for each page, as well as the exact location of all ROIs on the recognized pages. This is especially useful if you have fields with no bounding boxes, so on the EFI it is very hard to know exactly what size and location to give to the ROI. You can see from the results if there are any problems with identification, and adjust the EFI regions and parameters, if necessary.

1. In the Working Area, click to select the EFI.

2. Right-click the selected EFI to open the context menu and select **Test EFI**.

3. Select the batch file containing the TIFs for the forms you want to compare to the EFI and click **Open**.
    The **FormOut Test** dialog box opens. Testing begins immediately.
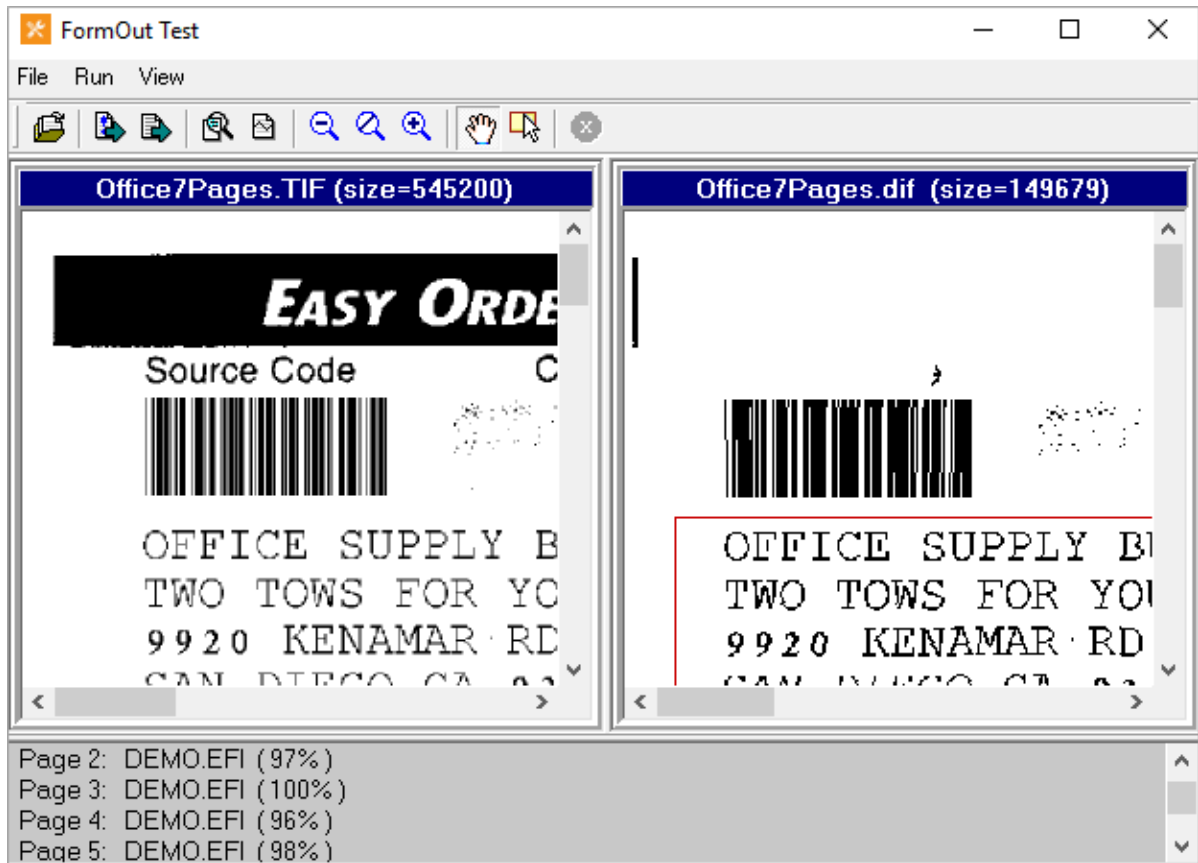
The image of the scanned test form – the TIF file – is displayed on the left. If the EFI is well-defined and eFlow was able to match it and remove it from the scanned image, the net image - the DIF file - is displayed on the right. The results area at the bottom of the screen shows a page-by-page listing with the EFI name and the quality of matching (in %) for each page scanned. Anything above 80% is considered a good match. Recognition of 97-100% is considered optimal.

If the EFI is not sufficiently well defined, the results area shows the error message: FRM could not be matched with EFI. In this case, and in the case of poor or unsatisfactory matching, you must rescan the EFI and/or adjust its parameters.

The **FormOut** Test dialog box contains the following buttons:

Button	Description
	Open an image batch file for testing.
	Open an EFI file for testing.
	Run the current batch of images. Used to re-run the test batch after rescanning the EFI and/or adjusting the EFI parameters.
	Run the selected page. Used to re-run selected pages of the test batch after rescanning the EFI and/or fine-tuning the EFI parameters. You can select individual pages from the results area at the bottom of the screen.

Button	Description
	Click this button to display the EFI or the DIF.
	If the DIF is currently displayed, click this button to view the EFI selected for comparison. If the EFI is already displayed when you click this button, the DIF is shown as it appears after FormOut!
	Resize the image: zoom out, return to the original size, or zoom in.

ⓘ During the test run, a **Stop** button appears, enabling you to interrupt the process.

Chapter 9

Define exceptions

Almost all forms have rules that apply to filled contents of the various fields in the forms. After you define these rules, the Validate station operator cannot leave the data completion process until these rules are obeyed. But what happens when the form was not filled in correctly? The operator cannot change the data of the form to match the predefined rules.

For these cases, eFlow allows you to define exceptions. In the Validate station, the operator can click the **Exceptions** button and select the most appropriate exception from the list, leaving the invalid data in the fields. The operator can then move on, and the exceptions are transmitted to a supervisor in the Escalate station for resolution.

Exception categories group exceptions. They are used to sort exceptions for workflow routing and other purposes. For example, you can filter the forms sent to a specific workflow module by specifying that only forms with field exceptions belonging to a specific exception category should be transferred.

To create exceptions for the application, you must first create appropriate exception categories.
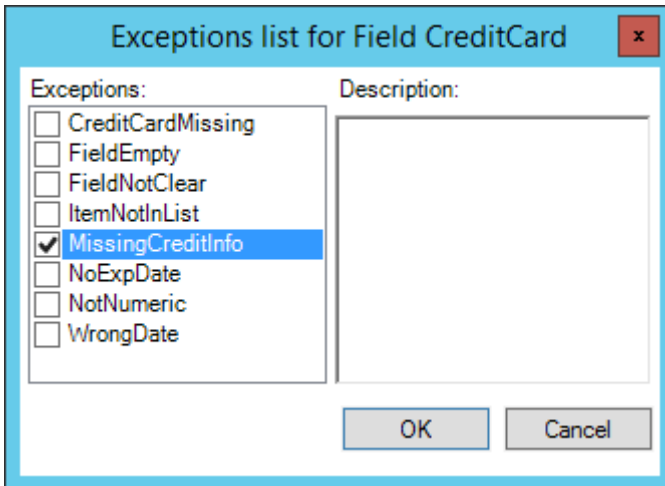
Create an exception category

Exception categories can only be added to the application object.

1. In the Application Explorer tree, click on the application object.

2. Right-click the highlighted object to open the context menu and select **New Exception Category**.

   The new exception category appears in the application tree. The Object Inspector displays the properties and events.

3. Set the exception category properties.

Exception category properties

Property	Description
Name	The exception category name. This name appears automatically in the Description box in the Object Inspector window after you create an exception category. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the exception category.
Caption	A name for the exception category that appears on the screen throughout the application.
Color	A color for the exception category, used for display purposes, as viewed by the supervisor in the Escalate station.

Create an exception

Exceptions can only be added to the exception category object.

1. In the Application Explorer window, click to select an exception category object.

2. Right-click the highlighted object to open the context menu and select **New Exception**.

   The new exception appears under the exception category. The Object Inspector displays the properties and events.

3. Set the exception properties.

Exception properties

Property	Description
Name	Name of the exception. This name appears automatically in the Description box in the Object Inspector window after you create an exception. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the exception.
Caption	A name for the exception that appears on the screen throughout the application.
Always Available	Select this check box to specify that the exception will be available to all fields/cases/forms in the Validate and Escalate stations.

Property	Description
Selected Action	Specify which action is to be performed when the field is tagged with an exception: • **Check the rest of the rules**: Continue to check the field for compliance with other rules. Mark the exception and move on. • **No check in the current field**: Do not continue to check the field for compliance with other rules. Disregard the whole field. • **No check in the current group**: Discontinue checking for compliance with rules for the entire field group. Disregard the whole field group. • **No check in the current form**: Discontinue checking for compliance with rules for the entire form. Disregard the whole form. • **No check in the current collection**: Discontinue checking for compliance with rules for the entire collection. Disregard the whole collection.
Hot Key	Specify which key combination will be associated with a specific exception. This hot key is used in the Validate and Escalate stations to assign an exception to a field or column.

Link exceptions to a field

For exceptions to perform their tasks they must first be linked to the fields for which they will be available.

> ⓘ There are two ways to link exceptions. Choose either option A or option B.

1. In the Application Explorer tree, click to select the field object.
   The Object Inspector displays the properties and events.

2. Select the exceptions:
   • Option A: Expand **Exceptions**. In the dropdown list for each exception, select **True**.

⊟ Exceptions	[FieldEmpty,FieldNotClear,ItemNotInList,NotNumeric,WrongDate
FieldEmpty	True
FieldNotClear	True
NotNumeric	True
BarcodeMissir	False
CreditCardMis:	False

   • Option B: Click ⊡ to the right of the **Exceptions** property. In the **Exceptions** dialog box, select the check box for each exception, then click **OK**.
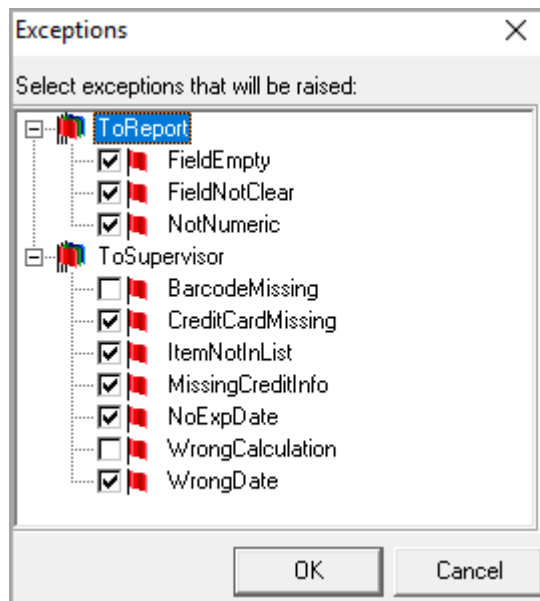
The selected exceptions are automatically set to **True** in the Object Inspector.

Define lookup tables

Lookup tables facilitate data entry in the Validate station by providing a selection list of possible field values. Known information, such as names, addresses, and identification numbers, can serve as predefined values that operators can select to replace unrecognized data. Values entered by an operator can be validated against the list of permissible values. Each item in the lookup table is associated with an appropriate field in a corresponding form.

The values of lookup tables may be stored in the following ways, depending on the type of lookup table:

- Directly in eFlow Design
- In a simple CSV text file
- In a database

Each item (list of values) in a lookup table must be assigned to the field to which it relates. For example, you would assign a list of telephone area codes to the **Phone Area** field. You can include more than one value in a lookup table. For example, you can define two lookup items in a table of items ordered by a customer: **Number** and **Description**. You would assign the **Number** lookup item to the **Item Number** field, and the **Description** lookup item to the **Item Description** field.

Create a lookup table

Lookup tables can only be added to the application object.

1. In the Application Explorer tree, click on the application object.

2. Right-click the highlighted object to open the context menu and select **New Lookup Table**.
   The lookup table object appears in the Application Explorer tree.

3. Type a name and press Enter.
   A new lookup item, **NewLI**, is automatically added under the lookup table. You can change the name of this item in the lookup item properties to reflect the actual content.

4. To add a new lookup item, right-click on the lookup table object and select **New Lookup Item**.

5. In the Application Explorer tree, click on the lookup table object. The Object Inspector displays the properties and events.

6. In the lookup table properties, select a lookup table **Type** and click ... to the right of the **Definitions** property to define the lookup table parameters. The parameters that are available depends on the table type.

Lookup table properties

Property	Description
Name	The name of the lookup table.
Description	A short description of the lookup table.
Caption	A name for the lookup table that appears on the screen throughout the application.
Type	Select the type of lookup table you want to create.
Definitions	Click ... to open a dialog box in which you can define the lookup table parameters.

Exorbyte parameters

With this lookup table type, you can store the lookup item values on an Exorbyte server.

Parameter	Description
Host Name	The name of the Exorbyte server.
Port Number	The port number of the Exorbyte server.

Parameter	Description
Min. Confidence	
Max. Distance	
Page size	The maximum number of records to be stored in memory.
Items	In this parameter, you map the lookup table items to columns in a file or database. • Mode: • Weight: • Column: • Name:

MS SQL parameters

With this lookup table type, you can store the lookup item values in a Microsoft SQL Server database.

Parameter	Description
Page size	The maximum number of records to be stored in memory.
Connection string	The name and location of the SQL database.
Table name	The name of the SQL table containing the lookup item values.
Items	In this parameter, you map the lookup table items to columns in a file or database. • Column: • Name:

Fuzzy parameters

With this lookup table type, you can store the lookup item values in a text file.

Parameter	Description
Case sensitive	Specify whether the lookup item values are case sensitive. For example, if this parameter is set to True and the lookup item contains the value **ABC**, the system will not accept the value **abc** if this is entered by the Validate station operator.
File Name	The name of the file containing the lookup item values.
Min. Confidence	
Items	In this parameter, you map the lookup table items to columns in a file or database. • SearchMode: Fuzzy or Complete • Column: • Name:

Fuzzy Server parameters

With this lookup table type, you can store the lookup item values on a Fuzzy Server.

Parameter	Description
Case sensitive	Specify whether the lookup item values are case sensitive. For example, if this parameter is set to True and the lookup item contains the value "ABC", the system will not accept the value "abc" if this is entered by the Validate station operator.
Service URL	The URL of the Fuzzy Server service.
Table name	The name of the Fuzzy Server table containing the lookup item values.
Items	In this parameter, you map the lookup table items to columns in a file or database. • SearchMode: Fuzzy or Complete • Column: • Name:

Legacy parameters

With this lookup table type, you can store the lookup item values in a text file, or enter them directly.

Parameter	Description
Case sensitive	Specify whether the lookup item values are case sensitive. For example, if this parameter is set to True and the lookup item contains the value "ABC", the system will not accept the value "abc" if this is entered by the Validate station operator.
Text data file	The name of the file containing the lookup item values.
Quick Text Data	Click the arrow to the right of this parameter to enter lookup item values.

Parameter	Description
Items	In this parameter, you map the lookup table items to columns in a file or database. • **Type**: • **Length**: • **Name**:
Addition Columns	

Chapter 11

Roles

This chapter details how different system-level and application-level roles affect the behaviors of the Tools, Reports and Stations modules. In the Design tool you can add and edit a role and set different permissions for the role. Behaviors of Tools, Reports and Stations modules change according the permission that is set in the Design tool.

General

The following table shows the Roles and their associated behaviors, which do not require any extra permissions in Design tool as compared to the roles mentioned in Tools, Stations and Reports.

Role	Behaviors
System: Administrator **Application:** Administrator	Can perform all the operations for Tools, Reports and Stations modules.
System: Execute **Application:** Administrator	For all the Tools/Reports/Stations modules, you can perform all operations except the following: **Supervise**: Settings cannot be changed. **Administrate**: You can open it but cannot create/delete applications, cannot create auto run stations, and cannot open Statistics.
System: Execute and Write **Application:** Administrator	For all the Tools/Reports/Stations modules, you can perform all operations except the following: **Administrate**: You can open it, can create /delete applications, can create auto run stations, but cannot open Statistics.
System: Read **Application:** Administrator/User defined	Cannot open any Tools, Reports and Stations modules.
System: Write **Application:** Administrator/User defined	Cannot open any Tools, Reports and Stations modules.
System: Read and Write **Application:** Administrator/User defined	Cannot open any Tools, Reports and Stations modules.

Add or edit a Role

This section describes how to add or edit a role in Design tool.

1. To add a role, go to **eFlow LaunchPro** > **Design**, and then click **Application Design**.

2. In the left pane, expand **Security** > **Roles**.
3. Right-click the **Roles** and select **New Role**.
   A new role is added.
4. Type a name for the newly added role.

Tools

For all the tools modules to open, edit and save changes selection of the Read flag is mandatory.

The following table shows the behaviors of the Design tool with different flags combinations.

Type	Name	CONFIGURE	EXECUTE	READ	WRITE
FLOW	OfficeStore	☐	☐	☑	☐
STATION	efControl	☐	☐	☐	☐
STATION	efDesigner	☐	☑	☐	☑
STATION	efFreedomDesigner	☐	☐	☐	☐
STATION	efsSmartDesigner	☐	☐	☐	☐
STATION	efFreedomAnalyzer	☐	☐	☐	☐
STATION	efOCRAnalyzer	☐	☐	☐	☐
STATION	efRecognitionAnalyzer	☐	☐	☐	☐
STATION	Deliver	☐	☐	☐	☐
STATION	Escalate	☐	☐	☐	☐
STATION	Validate	☐	☐	☐	☐
STATION	Recognize	☐	☐	☐	☐
STATION	Scan	☐	☐	☐	☐
STATION	Collect	☐	☐	☐	☐
STATION	Run	☐	☐	☐	☐
STATION	Custom	☐	☐	☐	☐
SERVICE	Security	☐	☐	☐	☐
SERVICE	NewServices	☐	☐	☐	☐
SERVICE	Setup	☐	☐	☐	☑
SERVICE	Statistics	☐	☐	☐	☐

Role	Setup	EXECUTE	WRITE	Result
System: Administrator **Application**: User defined Or **System**: Execute **Application**: User defined Or **System**: Execute and Write **Application**: User defined				Cannot open the Design tool.
			✓	Cannot open the Design tool.
		✓		Can open the Design tool but cannot save any changes made to it.
		✓	✓	Can open the Design tool but cannot save if any change is made as the Setup flag is not selected.
	✓	✓		Can open the Design tool but cannot save if any change is made as WRITE flag is not selected.
	✓	✓	✓	Can open the Design tool and save the changes made to it.

The behaviors mentioned in the above table is also applicable to the Design Flow, Design Free, Design Smart tools also.

Administrate: In the Administrate tool, you can create or delete applications, auto run stations, but cannot save the statistics if the CONFIGURE flag is not selected for the corresponding application. You can save the statistics if the CONFIGURE flag is selected for Statistics. Selection of other flags are not necessary.

Type	Name	CONFIGURE	EXECUTE	READ	WRITE
FLOW	Invoices	☐	☐	☐	☐
STATION	efControl	☐	☐	☐	☐
STATION	efDesigner	☐	☐	☐	☐
STATION	efFreedomDesigner	☐	☐	☐	☐
STATION	efsSmartDesigner	☐	☐	☐	☐
STATION	efFreedomAnalyzer	☐	☐	☐	☐
STATION	efOCRAnalyzer	☐	☐	☐	☐
STATION	efRecognitionAnalyzer	☐	☐	☐	☐
STATION	Scan	☐	☐	☐	☐
STATION	Collect	☐	☐	☐	☐
STATION	Recognize	☐	☐	☐	☐
STATION	BuildFree	☐	☐	☐	☐
STATION	Validate	☐	☐	☐	☐
STATION	CollectFree	☐	☐	☐	☐
STATION	Escalate	☐	☐	☐	☐
STATION	Deliver	☐	☐	☐	☐
SERVICE	Security	☐	☐	☐	☐
SERVICE	NewServices	☐	☐	☐	☐
SERVICE	Setup	☐	☐	☐	☐
SERVICE	Statistics	✔	☐	☐	☐

View CSM: In the View CSM, you can open and do changes to the Station/Reports/Tools modules only if the corresponding flags are selected for an application in the Design tool.

Stations

To open and process collections in a station, such as Collect, Recognize, and Escalate, selection of the associated flags are necessary.

> 🛑 For all the stations to open or process collections, selection of the Read flag is mandatory.

The following table shows the behaviors of the Collect station with different flags combinations.

Type	Name	CONFIGURE	EXECUTE	READ	WRITE
FLOW	OfficeStore			✔	
STATION	efControl				
STATION	efDesigner				
STATION	efFreedomDesigner				
STATION	efsSmartDesigner				
STATION	efFreedomAnalyzer				
STATION	efOCRAnalyzer				
STATION	efRecognitionAnalyzer				
STATION	Deliver				
STATION	Escalate				
STATION	Validate				
STATION	Recognize				
STATION	Scan				
STATION	Collect	✔	✔		✔
STATION	Run				
STATION	Custom				
SERVICE	Security				
SERVICE	NewServices				
SERVICE	Setup				
SERVICE	Statistics				

> ℹ️ The CONFIGURE flag is only applicable for Collect, Scan, Tile and Deliver2ERP stations.

Role	CONFIGURE	EXECUTE	WRITE	Behavior
System: Administrator				Collect station is not available in the Station list.
Application: User defined			✓	Collect station is available in the Station list but cannot open it.
Or **System**: Execute		✓		Can open the Collect station but cannot process the collection.
Application: User defined		✓	✓	Can open the Collect station and process collections.
Or **System**: Execute and Write **Application**: User defined	✓	✓	✓	Can open the Collect station, process the collections, and configure the Collect station.

ⓘ The behaviors mentioned in the above table is also applicable to the Validate, Collect Free, Deliver, Organize, FormId, ManualId, Manual Classify, Discard, Dynamic T, Clean, QA stations.

Reports

🛑 For all the reports module to open them you must select the Read flag.

The following table shows the behaviors of the Analyze Free module with different flags combinations.

Type	Name	CONFIGURE	EXECUTE	READ	WRITE
FLOW	OfficeStore			✔	
STATION	efControl				
STATION	efDesigner				
STATION	efFreedomDesigner				
STATION	efsSmartDesigner				
STATION	efFreedomAnalyzer		✔		
STATION	efOCRAnalyzer				
STATION	efRecognitionAnalyzer				
STATION	Deliver				
STATION	Escalate				
STATION	Validate				
STATION	Recognize		✔		✔
STATION	Scan				
STATION	Collect				
STATION	Run				
STATION	Custom				
SERVICE	Security				
SERVICE	NewServices				
SERVICE	Setup				
SERVICE	Statistics				

Role	Recognize		EXECUTE	Behavior
	EXECUTE	WRITE		
System: Administrator **Application**: User defined Or **System**: Execute				Cannot open Analyze Free.
			✓	Can open but cannot run as Recognize (WRITE and EXECUTE) are not selected.
	✓		✓	Can open but cannot run as Recognize (WRITE) is not selected.
Application: User defined Or **System**: Execute and Write **Application**: User defined	✓	✓	✓	Can open and run the report.

The following table shows the behaviors of the Analyze Recognition module with different flags combinations.

Role	EXECUTE	Behavior
System: Administrator **Application**: User defined Or **System**: Execute **Application**: User defined Or **System**: Execute and Write **Application**: User defined		Cannot open the Analyze Recognition.
	✓	Can open and run the Analyze Recognition.

> ℹ The same behavior is applicable for the Test OCR Page and the View Learning Plan modules.

The following table shows the behaviors of the Control module with different flags combinations.

Role	EXECUTE	WRITE	Behavior
System: Administrator **Application**: User defined Or **System**: Execute **Application**: User defined Or **System**: Execute and Write **Application**: User defined			Cannot open the Control.
	✓		Can open the Control and kill or logout other stations but cannot delete/prioritize/move collections.
	✓	✓	Can open the Control and can kill or logout other stations and can delete/prioritize/move collections.

For the OCR Page PRD view and the Supervise modules, no flags are required for them to open and run.

For the **Supervise** module, for the role **System**: Execute and **Application**: User defined/ Administrator, you can open the Supervise module but cannot modify the settings.

Chapter 12

Define rules and validation functions

Almost every form has rules concerning the validity of data in the form (such as minimum or maximum values, mathematical relationships between fields, and so on). These validation rules are defined during the setup process. These rules are especially important, as data entered in the fields during the data completion process must also obey these rules. The Validate station will prevent an operator from proceeding to the other fields until the contents of a field are corrected or, when the data is invalid and cannot be corrected, the operator marks an exception.

Some examples of rules:

- The field must not be empty
- The field must contain only digits
- The field must contain a valid date
- The field value is the sum of two other fields

Rules and validation functions are similar.

Rules are simple statements that are constructed using a virtual keyboard with alphanumeric characters and Boolean symbols. For example, a rule might stipulate: PhoneAreaCode=FaxAreaCode.

Validation functions are more complex and allow more sophisticated checks. They are written in Visual Basic code.
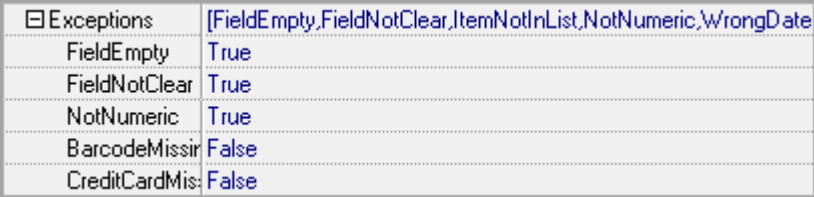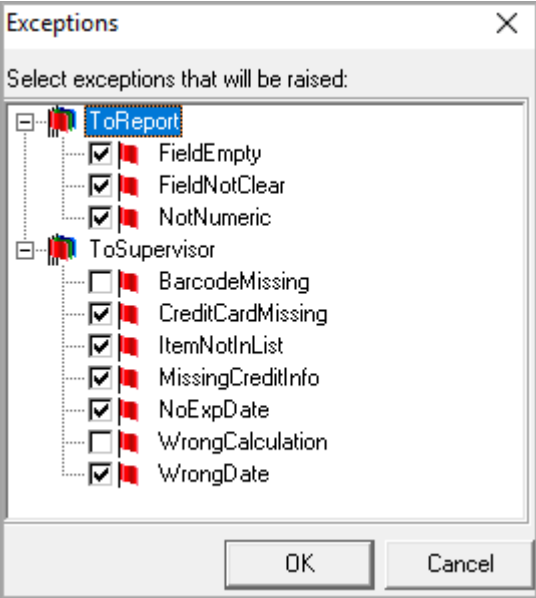
Create a rule

Rules can only be added to a field object or a table item object.

1. In the Application Explorer tree, click to select the field or table item object.

2. Right-click the highlighted object to open the context menu and select **New Rule**.

   A new rule object appears under the field or table item. The Object Inspector displays the properties and events.

3. Type a name and press Enter.

4. Set the rule properties.

Rule properties

Property	Description
Name	The name of the rule. This name appears automatically in the Description box in the Object Inspector window after you create a rule. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the rule.
Caption	A name for the rule that appears on the screen throughout the application.
RuleString	A string that defines the rule. Rule strings are constructed of field values, constants, and operators. You can build rules with the Rules Editor. Click ⋯ to the right of the RuleString property to open the Rules Editor dialog box. See Use the Rules Editor for more information.
RuleMessage	The error message displayed in the Validate station when the data entered by the operator does not conform to the rule.

Property	Description
Exceptions	Specify the available exceptions for the rule. Exceptions allow Validate station operators to leave invalid data in a field if they are unable to correct it. The exceptions can be resolved by a supervisor in the Escalate station. See Define exceptions for more information. There are two ways to link exceptions. Choose either option A or option B. • Option A: Click the + button to the left of the **Exceptions** property. In the dropdown list for each exception, select **True**. • Option B: Click ... to the right of the **Exceptions** property. In the **Exceptions** dialog box, select the check box for each exception, then click **OK**. The selected exceptions are automatically set to True in the Object Inspector.
RuleAction	Specify what the system should do if the conditions of the rule are not met. • **Show Message and Stop**: Display an error message in the status bar, but do not move to the next field. The operator must assign an exception. • **Attach Exceptions and Continue**: Automatically assign the rule's exceptions and move to the next field.

Use the Rules Editor

The Rules Editor enables you to define rules quickly and easily. These rules are used to validate the data and are applied to fields in the same form.

1. In the Object Inspector, click ⊡ to the right of the **RuleString** property.

2. In the **Rules Editor** dialog box, enter the rule in the rule string box.
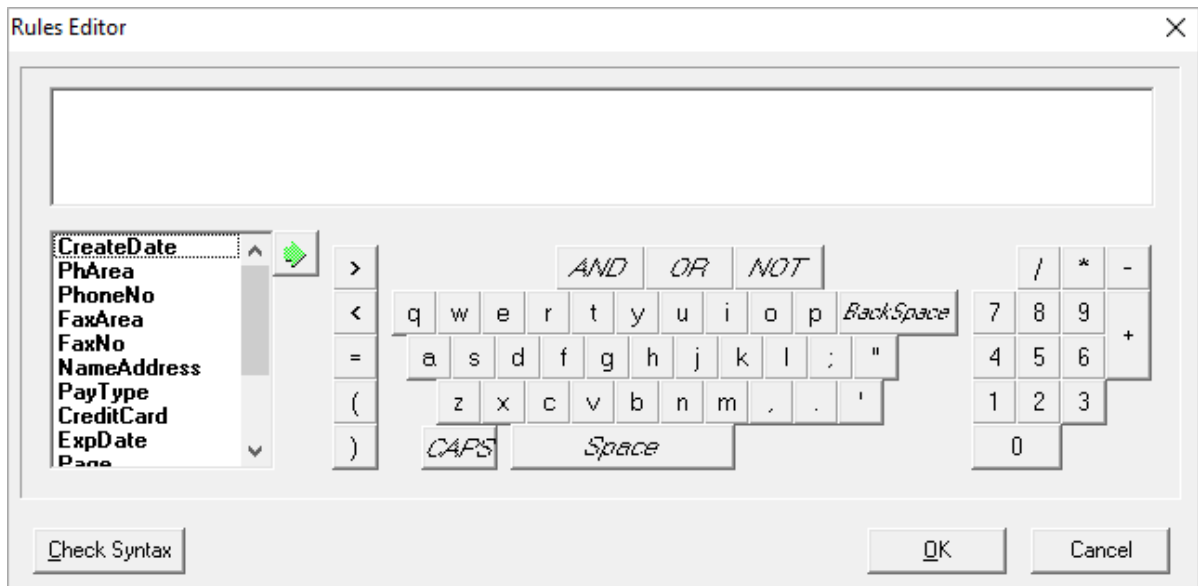
3. To add a field to the rule string, double-click on the field name in the fields list box on the left of the dialog, or select the field and click **Add To Rule** ➡.

4. The **Rules Editor** dialog box has logical operator buttons, an alphabetic and numeric keypad, and buttons representing the Boolean expressions AND, OR and NOT. Click the corresponding button or key to add operators, characters, numbers, and expressions to the rule string.

5. When you have finished building the rule, click **OK**.

    The rule syntax is added to the **RuleString** property.

Rule expression syntax

There are two types of expression syntax:

*   Mathematical expressions
*   Field format expressions

The **Field Type** property determines whether the rule will be written as a mathematical expression or a text string (field format). For example, dates and numbers take mathematical rules, and text fields take field format rules.

Mathematical expressions

Types

Two types of mathematical expression are supported:

- Numbers
- Dates

The number type is used for integers and real numbers. The date type is used for dates. You can only compare numbers with numbers and dates with dates (not numbers with dates).

Constants

If you type numbers or dates in the rule, these remain constant even as the field values change. For example, you can specify that a field will always have a given value, such as AreaCode=212.

The supported constants are:

- Number constants
- Any number (syntax: with or without a decimal point).

Variables

A variable is a field name or column name, and the value of the variable is the content of the field or column.

Syntax: idnumber. The table is the active table.

> ℹ If you set one column equal to another column, every cell in the first column must be equal to the corresponding cell in the second column. Similarly, if you specify that Column 1 is greater than Column 2, every cell in Column 1 must be greater than the corresponding cell in the Column 2.

Operators

Category	Operator	Description
Numerical	+	Addition
	-	Subtraction
	*	Multiplication
	/	Division
Operators	>	Greater than
	<	Less than
	=	Equal to
	>=	Greater than or equal to
	<=	Less than or equal to
	<>	Between
Boolean	AND	Logical AND
	OR	Logical OR
	NOT	Not equal to

Field format expressions

The following are examples of field format expressions:

Telephone number

A telephone number may appear as a three-digit prefix followed by a hyphen and a seven-digit number.

The syntax is as follows:

000-0000000

Credit card number

A credit card number may appear as a sixteen-digit number in four groups of four digits separated by a space.

The syntax is as follows:

0000 0000 0000 0000

Vehicle license plate number

A vehicle license plate number may appear as two alphanumeric characters followed by a dash, two required digits, an optional digit (represented by the 9 character), another dash, and two more alphanumeric characters.

The syntax is as follows:

CC-009-CC

Validation functions

Validation checks allow for sophisticated, automatic checking of field contents at many different levels of inquiry. Not only can fields be related, but they may depend on one another. For example, zip codes, city names, and state names are related. Similarly, telephone area codes and city names have a certain relationship since telephones area codes are geographically determined. Date fields may be logically limited to certain dates, with any other dates being considered invalid. For example, if in a current census, a respondent answered that he was born in 1840, the date field could not be considered valid. Validation functions can be designed to perform these sorts of checks.

Create a validation function

Validation functions can only be added to a field object or table item object.
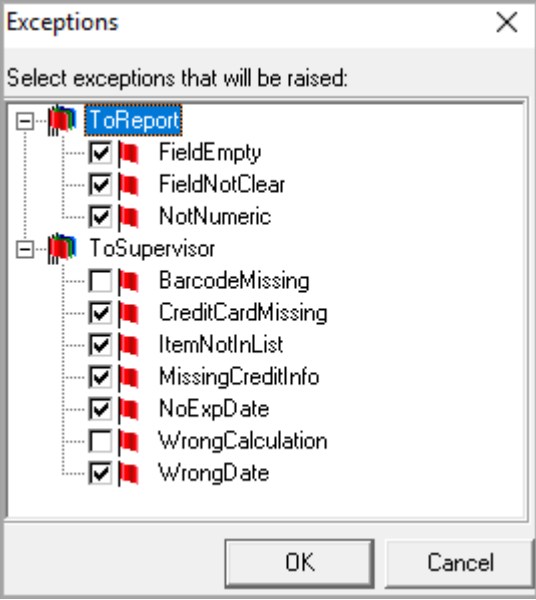
1. Define a customized validation function using a DLL. The DLL can be placed in the eFlow Bin folder or anywhere else in the system search path.

2. In the Application Explorer tree, click to select the field or table item object.

3. Right-click the highlighted object to open the context menu and select **New VFunction**.

A new validation function object appears under the field or table item object. The Object Inspector displays the properties and events.

4. Set the validation function properties.

5. In the **Events** tab, in the **OnValidate** field, type in the function or method, or click ▭ to select an assembly file. See Event functions for more information.

Validation function properties

Property	Description
Name	The name of the validation function. This name appears automatically in the **Description** box in the Object Inspector after you create a validation function. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the validation function.
Caption	A name for the validation function that appears on the screen throughout the application.
RuleMessage	The error message displayed in the Validate station when the data entered by the operator does not conform to the rule.

Property	Description
Exceptions	Specify the available exceptions for the validation function. Exceptions allow Validate station operators to leave invalid data in a field if they are unable to correct it. The exceptions can be resolved by a supervisor in the Escalate station. See Define exceptions for more information. There are two ways to link exceptions. Choose either option A or option B. • Option A: Expand **Exceptions**. In the dropdown list for each exception, select **True**. • Option B: Click ⋯ to the right of the **Exceptions** property. In the **Exceptions** dialog box, select the check box for each exception, then click **OK**. The selected exceptions are automatically set to True in the Object Inspector.
RuleAction	Specify what the system should do if the conditions of the rule are not met. • Show Message and Stop: Display an error message in the status bar, but do not move to the next field. The operator must assign an exception. • Attach Exceptions and Continue: Automatically assign the rule's exceptions and move to the next field.

Chapter 13

Design screen layout

The eFlow Layout Designer enables you to customize the screen layout of the Validate station. The Layout Designer is integrated in the eFlow Design module.

When you create a layout, you determine which fields, tables, and other objects (such as a picture or an image viewer) to display, and the position, order, appearance and behavior of the objects. You can quickly and easily add objects to the layout by dragging and dropping them into the layout design area. Toolbar buttons provide functions for precise positioning of the objects.

You can create layouts for forms or field groups. A form layout displays all fields of an entire form on the screen. A group layout displays the same field or group of fields from successive forms. You can also create different layouts for the thick client Validate station and the browser-based Web Validate station.

This section briefly explains how to create a Validate station layout.

Open the Layout Designer

1. In the Application Explorer, click on a form or field group object.
2. Right-click the highlighted object to open the context menu and select **Edit Screen Layout**.

The Layout Designer desktop consists of the following areas.

ⓘ Most of these areas are empty until you select a layout in the **Layouts tree**.
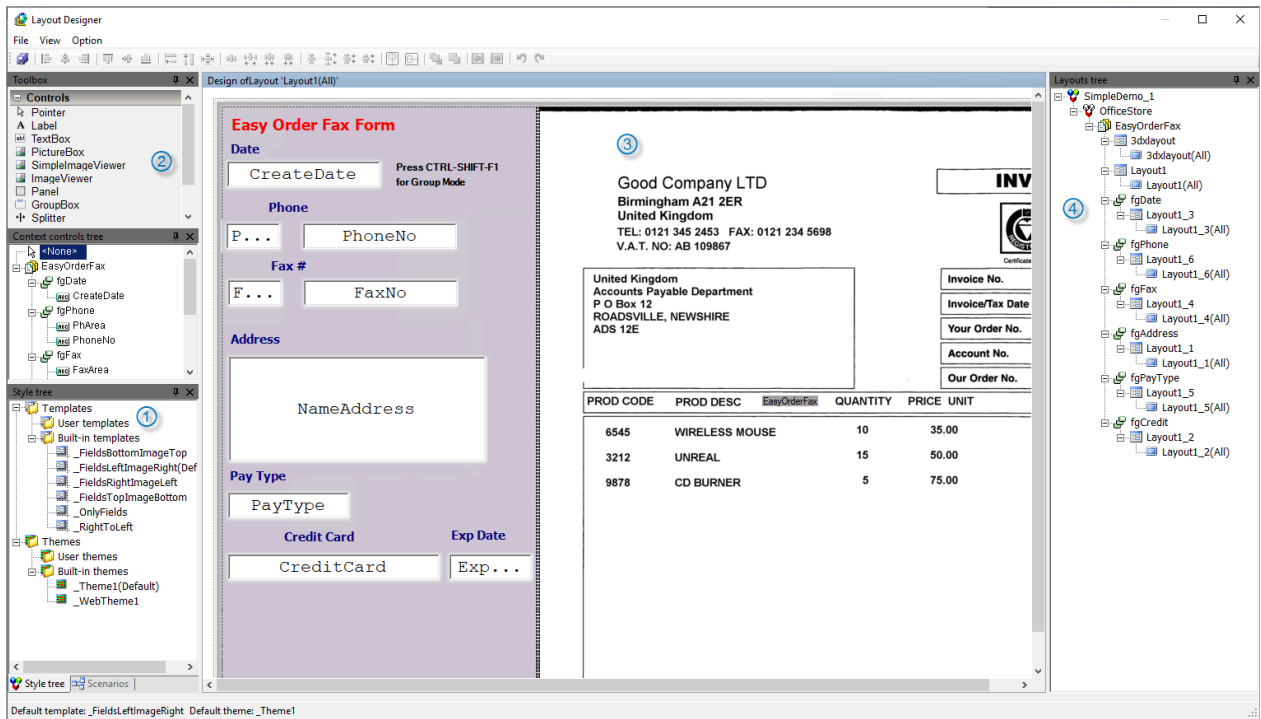
1	**Style tree**: Contains all the templates and themes that you can add to your layout.
2	**Toolbox - context control tree**: Contains all objects (forms, groups, fields and tables) that have been defined for the eFlow application in the Design module.
3	**Design of layout**: Contains all fields and other objects that will be displayed to operators in the Validate station.
4	**Layout tree**: Contains all layouts that have been defined for the eFlow application.

Create a layout

1. In the **Layouts tree**, right-click on a form or group and select **New Completion Layout** or **New Web completion Layout**.

   A new layout appears in the tree.

2. Right-click on the new layout and select **Rename**.

3. Type a name and press Enter.
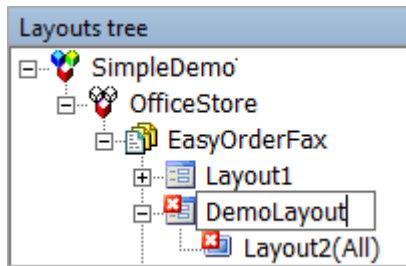
You must now add controls to the layout.

Add controls to a layout

The quickest and easiest way to add controls to a layout is to drag the relevant objects from the **Context controls tree** to the **Design of Layout** area.

For example, if you select the form and drag it to the **Design of Layout** area, all objects (forms, groups, fields and tables) that have been defined for the eFlow application in the Design module are added to the layout.

1. In the **Context controls** tree tab, click on the form or field group. Holding down the left mouse button, drag the form or group into the **Design of Layout** area, then release the mouse button.

    All fields and other objects, such as an image viewer, appear in the **Design of Layout** area.
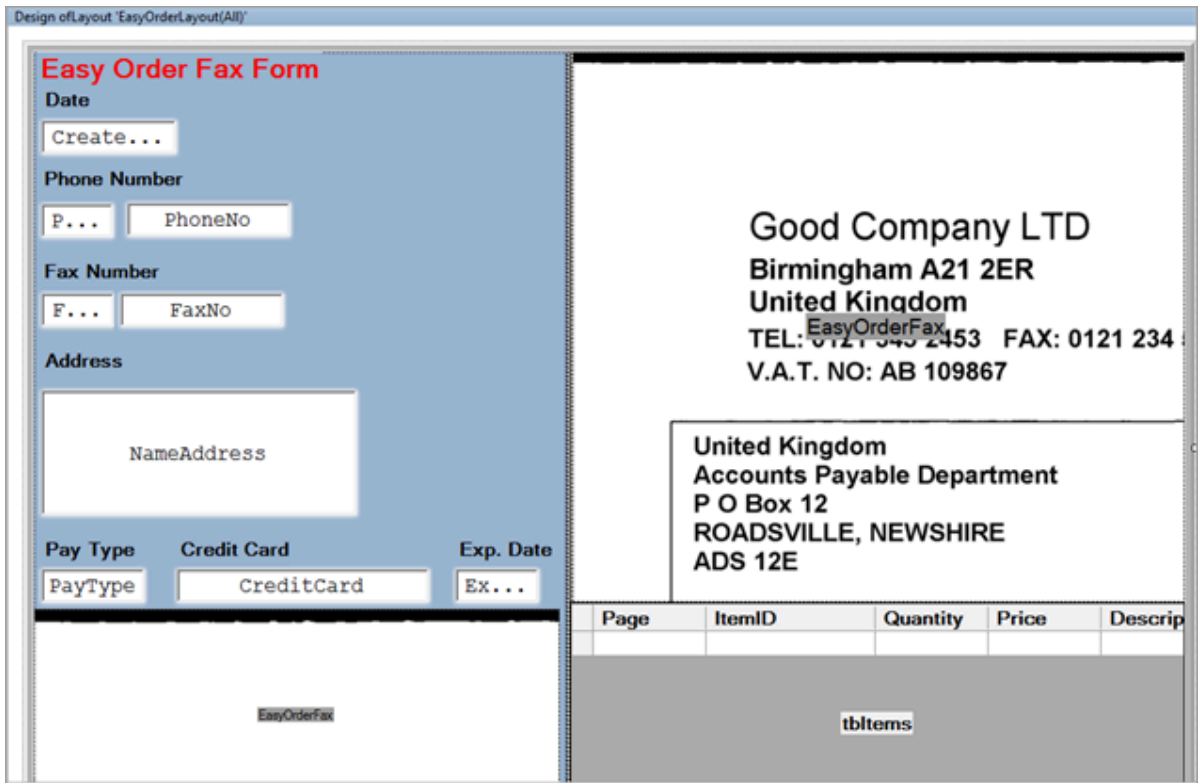
2. If necessary, in the **Design of Layout** area, change the size of the layout by dragging its borders with the mouse.

3. Click **Save All** ![icon] on the toolbar.

Chapter 14

Design OCR engine

Forms (paper or scanned images) contain fields with characters, letters, punctuation symbols, barcodes, and numbers that must be recognized and processed to be converted into electronic data. OCR engines perform these tasks. OCR engines are programs that are specifically created to recognize, process, and convert specific types of data, whether barcodes, handwritten numbers, or machine print characters.

The strongest and most powerful of the many different OCR and ICR engines provided by leading companies in the recognition field are available to you through eFlow. OCR (Optical Character Reader) engines are used for machine print, and ICR (Intelligent Character Reader) engines are used for handwritten text.

The OCR Engines Designer provides pre-tuned engine templates that can be used for character recognition. It also provides sophisticated parameter control, fine-tuning, and recognition results generation and checking for the OCR engines that you use.

Each engine contains a set of parameters defining how recognition is performed for a particular field. Each engine can be fine-tuned using the many parameter settings.

The OCR engine tester allows for immediate checking of results, and more accurate fine-tuning with greater control over engine settings. Settings can be checked immediately using the engine tester, which instantly processes results for selected fields.

OCR Engines Designer desktop

You select and configure OCR engines on the OCR Engines Designer desktop. To open the OCR Engines Designer desktop, on the Application Designer desktop toolbar, click **OCR Engines Designer** .
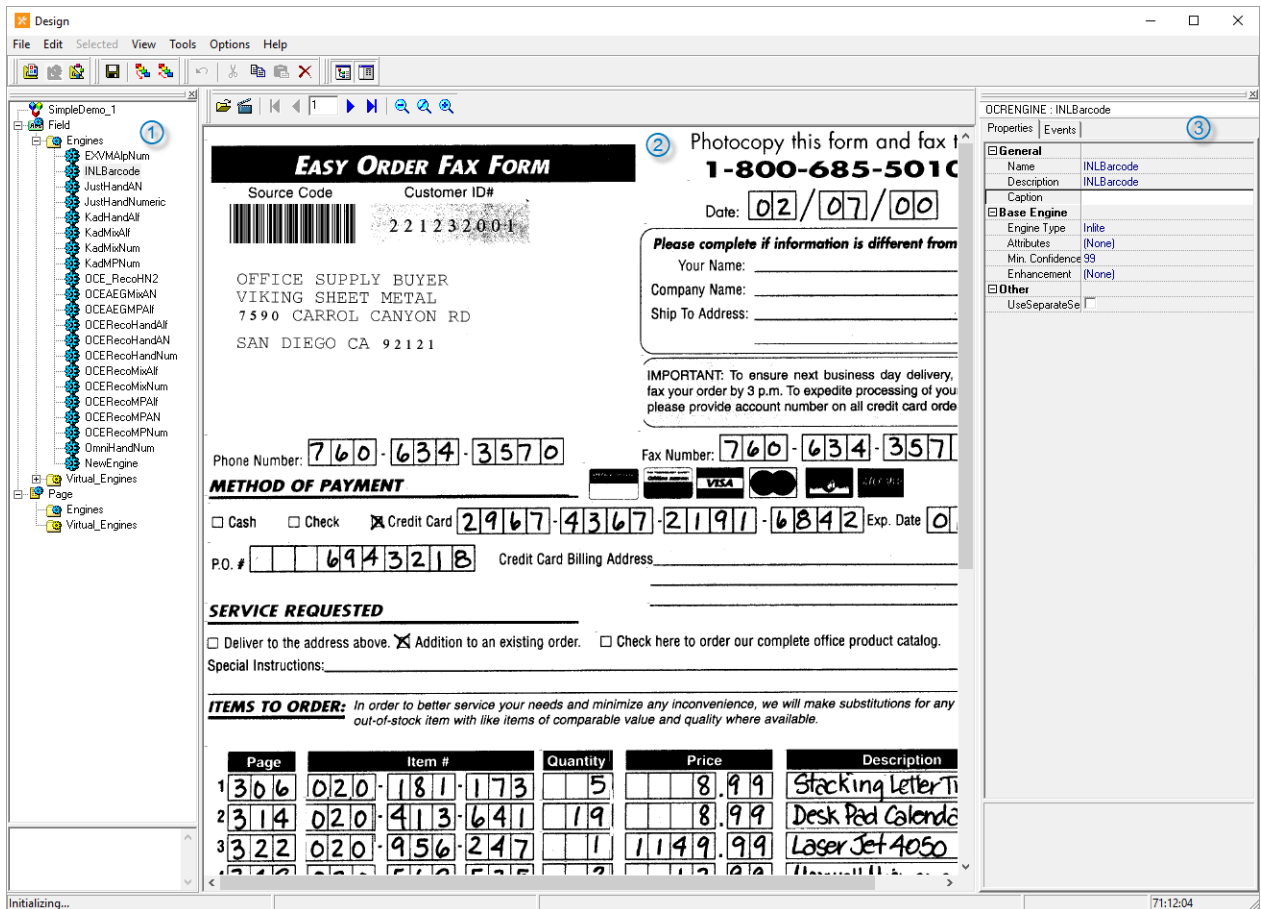
The OCR Engines Designer desktop consists of three primary sections:

1	OCR explorer
2	Working area
3	Object inspector

Create an OCR field or page engine object

1. In the OCR Explorer, under **Field** node or **Page** node, right-click **Engines** and on the context menu, select **New Engine**.

   A new engine object appears in the engine object tree. User created engine objects are symbolized by the engine object icon ![icon]. The Object Inspector displays the properties and events.

2. Type a name and press Enter.

3. Set the OCR engine properties.

ⓘ When you set values for an engine, these values will be set for all copies of the engine as well.

General properties

Property	Description
Name	The name of the engine. The name should be alphanumeric. This name appears automatically in the Description box in the Object Inspector after you create an engine. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the OCR engine.
Caption	A name for the engine that appears on the screen throughout the application.

Base engine properties

This section describes the Base engine properties.

Engine type

Several different engine types are available in the Engine Type selection list.

OCR field engine types

OCR field engines recognize the content of fields based on the recognized EFI or script.

The OCR field engine types include:
- Ceyes (Ligature)
- Character
- ExperVision TypeReader®
- Gentriqs
- Inlite Research Barcodes
- KADMOS™
- NestorReader™
- OCEAEG
- OCEReco
- Parascript
- RecoStar

OCR page engine types

OCR page engines recognize all the words in the image to help script recognition.
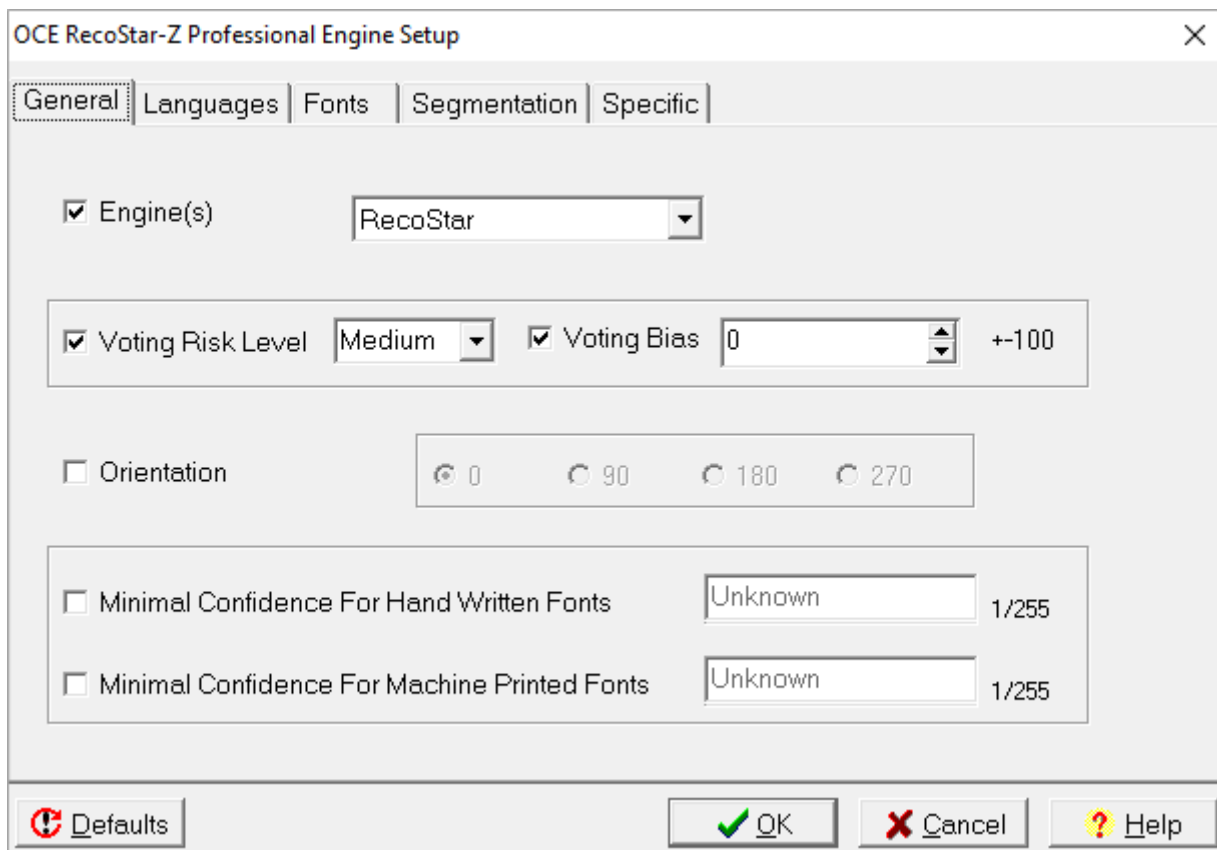
The OCR page engine types include:

- ExperVision
- Oce
- Omnipage
- Ricoh
- Abbyy

> ⓘ The Abbyy engine will be deprecated in a future Kofax eFlow release.

Attributes

Each engine type has its own specific set of parameters that can be set and fine-tuned so that the best possible recognition can be achieved. An engine is a combination of the engine type and the engine specific parameters. You can set the parameters in the **Engine Attributes** dialog box. To open this dialog box, click ⋯ to the right of the Attributes parameter.

> ⓘ The engine attributes dialogs are subject to change, as eFlow incorporates engine upgrades.

The main parameters that appear in many engines are the following:

- Language/Classifiers: Select the natural human language (such as English or French) and the type of character (such as hand-print or machine-print).
- Mask: Set the allowed characters, such as A-Z a-z (alphabetic) and 0-9 (numeric). Special characters, such as & or @, can also be defined.
- Character size/height: Set minimum/maximum values for character size.
- Distance between characters.
- Grid removal, noise processing, and enhancement.

The **Defaults** button restores the engine default settings.

Minimal confidence

Enter a number in the **Min. Confidence** field to specify the minimal confidence percentage that a resolved character must have for the result to be accepted. After the engine recognition algorithms have run and each result has been assigned a confidence value, results with lower confidence values than the specified value will be rejected. The default value is different for each engine and is related to general engine behavior.

Enhancement

Enhancement enables post-scan image modification. Click ▦ to specify the image enhancement to be applied before an engine performs recognition.

The following options are available:

- **Deskew** to correct alignment
- **Noise removal** to remove different kinds of noise, such as dots and specks
- **Lines** to remove vertical or horizontal lines
- **Character enhancement** to improve character recognition
- **Rotation** to correct an image that was rotated when it was scanned

You can perform enhancement in more than one place in the flow. In each instance, you can combine elements of the different enhancement filters. See The Enhancement Builder for more information.

Other properties

You can use the recognition of one engine (that is, how it reads characters) and the segmentation of another engine (that is, how it separates characters). Select the **UseSeparateSegmentation** check box to select a different engine for segmentation.

A new set of properties for the segmentation engine is displayed.

Segmentation properties

You can define one engine to recognize character data, and another engine to define the segmentation of data.

Segmentation is how the OCR separates characters. The segmentation algorithm divides the region into different segments. The segmentation mode specifies a treatment for the region. Using different recognition techniques, the segmentation algorithm divides the area into a minimal number of segments, a normal number of segments, or a maximal number of segments.

For example, the segmentation algorithm can ignore small gaps. So, if a region contains a B with a slight gap in it, what might be read as 13 using maximal segmentation is read as a B using minimal segmentation.

Minimal segmentation: read as B

Maximal segmentation: read as 13

If you choose a separate segmentation engine (by selecting the **UseSeparateSegmentation** check box), you can set the following properties for both OCR field and page engines to further improve recognition:

Property	Description
Engine type	Select the segmentation engine type.
Attributes	Each engine type has its own specific set of parameters that can be set and fine-tuned so that the best possible recognition can be achieved. An engine is a combination of the engine type and the engine specific parameters. You set the parameters in the **Engine Attributes** dialog box. To open this dialog box, click ... to the right of the Attributes parameter.
Min. Confidence	Enter a number in the **Min. Confidence** field to specify the minimal confidence percentage that a resolved character must have for the result to be accepted. After the engine recognition algorithms have run and each result has been assigned a confidence value, results with lower confidence values than the specified value will be rejected. The default value is different for each engine and is related to general engine behavior.
Enhancement	Enhancement enables post-scan image modification. Click ... to specify the image enhancement to be applied before an engine performs recognition.
	See The Enhancement Builder for more information.

Additional OCR page engine properties

In addition, the following properties are available for OCR page engines:

Property	Description
Use Clean Process	Select this check box to execute another noise removal algorithm. 1. Black on white areas are converted to white on black. 2. Dot matrix character recognition is improved.
Hand Print	Select this check box for pages that include hand-printed text.
Machine Print	Select this check box for pages that include machine-printed text. This is the default setting.
ICR Engine	Select from the list of previously created virtual field engines.

Chapter 15

Design virtual engine

The eFlow platform employs the most advanced OCR and ICR algorithms to extract text data from the images. Many recognition engines are supplied with the system, each suitable for a certain kind of text or data.

To improve recognition results, instead of a single OCR engine, you can group engines together to create an OCR virtual engine. A virtual engine contains one or more engines, specifically configured to optimize the recognition capabilities. Each engine has specific strengths in recognition performance. When combining engines in a virtual engine, choose engines that are most suited to the type of data that will be found in the specified region.

Every OCR engine supplies a prediction of each character that it examines and supplies a confidence level for every character it recognizes. The combined results from the OCR virtual engine are merged using a voting method.

The OCR engine tester allows for immediate checking of results, and more accurate fine-tuning with greater control over engine settings. Settings can be checked immediately using the engine tester, which instantly processes results for selected fields.

Create a virtual engine

1. In the OCR Explorer tree, under the **Field** node or **Page** node, click **Virtual Engines**.

2. Click the right mouse button to open the context menu. Select **New Virtual Engine**.

    A new virtual engine object appears in the engine object tree. User-created virtual engines are symbolized by the virtual engine icon 🐜 . The Object Inspector displays the properties and events.

3. Type a name and press Enter.

4. Set the engine properties.

5. Attach OCR engines to the virtual engine and set the order of the engines.

Virtual engine properties

General properties

Property	Description
Name	The name of the engine. The name should be alphanumeric. This name appears automatically in the Description box in the Object Inspector after you create an engine. You can change the description if you want to add additional information that may be helpful to you.
Description	A short description of the OCR engine.
Caption	A name for the engine that appears on the screen throughout the application.

Recognition mask properties

ⓘ These properties apply only to OCR field engines, not to OCR page engines.

The recognition mask properties define the type of print and allowed characters and punctuation. Define the **Print Type** and **Field Type** to achieve the best recognition results.

Property	Description
Field Type	The expected type of field: alphabetic, numeric or mixed alphanumeric.
Print Type	The expected type of print: handwritten, machine printed, both handwritten and machine printed, or barcodes.
Chars Case	The expected character case: uppercase, lowercase, or mixed uppercase and lowercase.
Allowed Characters	The characters that are expected to appear in a field. Click ⋯ to open the **Character Map** dialog box, where you can specify a range of allowed characters. For example, you can define the recognition mask so that only uppercase and lowercase characters between A and E are allowed. In the **Character Map** dialog box, specify the following string: A-Ea-e Only those characters, such as Abba, will be allowed.

Property	Description
Allowed Punctuation	Select this check box if you want to be able to recognize punctuation, such as decimal points or commas between figures, or special characters, such as arithmetic signs. By default, the engines ignore punctuation. For example, you can define the recognition mask so that the commas and decimal point in the following number are recognized: 1,234,567.89

Segmentation properties

ⓘ These properties apply only to OCR field engines, not to OCR page engines. All these settings are per OCR virtual engine, which may be connected to one or more regions.

Segmentation properties define the size of characters and spaces.

Property	Description
Monospaced	Select this check box if all the characters in the field are of the same width.
Min. Height Max. Height	The minimum and maximum allowed height of the characters per ROI, in pixels. Characters that exceed the limit (smaller or larger) will be ignored.
Min. Width Max. Width	The minimum and maximum allowed width of the characters per ROI, in pixels. Characters that exceed the limit (smaller or larger) will be ignored.
Super Segmentation	When this check box is selected, eFlow employs the voting technique to the segmentation results of several engines. This provides better recognition because the coordinates are calculated for the best probability. ⓘ Please consult Kofax when implementing super segmentation.

Voting properties

To improve recognition, the combined results from several OCR engines can be merged using a voting method, which defines what the result of the virtual engine will be.

OCR page engines have only one voting property, the **Voting Method**. The only available voting method value is **XVoter**.

OCR field engines have the following voting properties.

Property	Description
Voting Method	The voting methods for OCR field engines are **Safe**, **Normal**, **Majority**, **Order**, and **Equalizer**. The default is **Normal**. See the example below for detailed information on these methods.
Max Rejected Chars	Set the minimum percentage of recognized characters out of the total number of characters that an engine must return to take part in the voting. If the engine recognizes less than this percentage, it does not take part in the voting. The default value is 100%, that is, an engine must recognize every character to participate in the voting.

Property	Description
Max Guesses	Set the maximum number of failed guesses before recognition fails. The default is 1.

The following example illustrates the different outcomes that can be obtained from a group of results from different engines.

An OCR virtual engine contains five engines, which return the following results for a certain field. The asterisk * indicates that the character was not recognized by the engine.

Real value: 253478

Engine	Result
1	25***8
2	2*5378
3	253478
4	2*34*8
5	25***8

The following table describes the algorithms for each voting method and shows the results that each algorithm would return in the above example.

Voting method	Description	Result
Safe	The result of the vote must be unanimous, that is, all engines must return the same result.	2****8
Normal	The voting is unanimous if there is no conflict. Only engines that return a result are counted.	25**78
Order	The first engine (according to the order within the OCR virtual engine) that is above the confidence threshold determines the result.	255378
Majority	This is a simple majority vote. If there is no majority, then the result is rejected.	253478
Equalizer	The result is normalized by an algorithm that uses lookup tables and weighted rankings to compare and improve engine recognition results. See The Equalizer for OCR field engines for more information.	253478

ⓘ For all voting methods, voting is performed character by character, according to the detected segmentation for the character. If, for any reason, an engine is not able to determine a specific character's correct segmentation, the voting for this character will ignore the specific engine result or lack of result.

Other properties

ⓘ These properties apply only to OCR field engines, not to OCR page engines.

Property	Description
Enhancement	Click ▪▪▪ to specify the image enhancement to be applied before an engine performs recognition. See The Enhancement Builder for more information.
Keep Enhancement	Select this check box to apply image enhancement to the DIF image file. To disable enhancement, clear the check box.

Attach an OCR engine

Attach engines to the virtual engine by copying and pasting them.

1. Click to select the engine object.
2. Right-click the highlighted object to open the context menu and select **Copy**.
3. Click to select the virtual engine to which you wish to attach the engine.
4. Right-click the highlighted virtual engine to open the context menu and select **Paste**.

A copy of the engine appears in the engine object tree. This copy is called an engine link, symbolized by the engine link icon 🔗.

> ℹ️ You can change the properties of a user-defined engine. When you set or change values for an engine, these values will be set for all engine links as well.

Order OCR engines within a virtual engine

eFlow gives you full flexibility to change the order in which engines are applied to a region. The order of the OCR engines in the OCR Explorer tree determines the order in which they are applied to the image.

1. Click to select the engine object.
2. Drag and drop the engine to wherever you wish to place it.

The position of the engine in the hierarchy changes. All engine properties are preserved.

The Equalizer for OCR field engines

> ℹ️ The Equalizer applies only to OCR field engines, not to OCR page engines.

The Equalizer is a voting method that can be used to improve recognition results by incorporating the Equalizer algorithms. These algorithms weigh the value of engine results according to confidence levels to improve the accuracy of these results. The Equalizer can be used with or without lookup tables.

Each engine provides a guess and assigns it a confidence level. Each engine's results must also pass a minimal segmentation overlap test for its guesses to be accepted. By default, this value is 70%, the recommended overlap value. If the engine segmentation does not pass this overlap test, the Equalizer algorithm does not use its guesses.

All engine results that pass the segmentation test are used and ranked by the Equalizer. The Equalizer algorithm has a table that assigns a rank to each engine's confidence levels. The engine rankings are based on the engine's ability to recognize information accurately. The Equalizer algorithm uses a hard-coded default table of ranks, but it is possible to assign an alternate, custom ranking system to the top engine guesses.

A result's final adjusted rank is determined in the following way. The ranks of identical guesses are added to produce a sum, which comprises an adjusted rank. These adjusted ranks are then compared, and the two highest ranked guesses are selected for further testing.

The two highest ranking guesses are subjected to three tests. The tests involve the properties **Min. First**, **Max. Second**, and **Min. Difference**. If these guesses do not pass all three tests, an unrecognized character is returned.

Voting method

The Equalizer is a sophisticated voting method. Each engine has a voting table. The Equalizer compares the confidence levels results according to the confidence tables of each engine. This process is done for each character separately.

Let us assume 3 engines. All have the same confidence table for all guesses. The string to be recognized is the word "Hello".

Confidence	0-20	20-50	50-70	70-101
Score	0	1	2	3

The following table shows the results and confidence percentages returned by each engine for each character.

Confidence	H	E	L	L	O
Engine 1	H (25%) E(80%)	E (67%)	L (83%)	L (20%)	O (99%)
Engine 2	H (95%)	E (29%) F (81%)	L (80%)	L (49%)	O (69%) 0 (71%)
Engine 3	H (100%)	E (75%)	L (49%)	L (17%) I (64%)	O (18%)

The following table shows the scores assigned based on the confidence table.

Score	H	E	L	L	O
Engine 1	H (1) E (3)	E (2)	L (3)	L (1)	O (3)
Engine 2	H (3)	E (1) F (3)	L (3)	L (2)	O (2) 0 (3)
Engine 3	H (3)	E (3)	L (1)	L (0) I (2)	O (0)
Total	H (7) E (3)	E (6) F (3)	L (7)	L (3) I (2)	O (5) 0 (3)

Confidence tables

The default values of confidence tables for first and other guesses:

Engine	Default values for first and other guesses
OCE	20,40,60,101
AEG	20,50,99,101
CGK	20,45,90,101
Charactell	20,35,75,101
Kadmos	20,70,85,101
Gentriqs	50,90,99,101
Nestor	95,97,99,101
Ligature	95,97,99,101
Others	20,40,85,101

Equalizer parameters

The Equalizer parameters are displayed when you select **Equalizer** voting method.

Common parameters

Use common parameters to configure the Equalizer algorithm.

Parameter	Description
Equalizer Type	Select the level of strictness of the Equalizer algorithm. • **Normal**: Accept the Equalizer results only when the best recognition result is returned by all the engines, even if it is returned with a low confidence. There is essentially no second choice. • **Majority**: Accept the best recognition result from the majority of engines. • **Safe**: Accept the results only when all the engines return the same recognition result as the first choice, with a high confidence level.
Mask	Specify a string field mask using the syntax A (alphabetic), N (numeric) and C (character). For example, to accept a field where the expected value is 2 alphabetic characters followed by 3 numeric characters, enter the following string in the box: AANNNN
Max. Second	Specify the maximum acceptable confidence level for a secondary guess.
Min. Difference	Specify the minimum acceptable confidence level difference between primary and secondary guesses.
Min. First	Specify the minimum acceptable confidence level for a primary guess.

Parameter	Description
Mapping	Specify a mapping string separated by semicolon. The first letter is the mapping character, the others are alternate characters to be mapped to first character.
	For example, if different A characters may appear (â, ä, æ, à, á, å) in a field, but only the A character is desired, enter the following string in the box:
	aâäæàáå
	Separate multiple strings with semicolons.
	ⓘ As many engines exist to recognize specialized character sets, string mapping is usually not necessary.
Overlap	Specify the minimum required overlap between the first engine and the rest of the engines. The recommended value is more than 50.

Dictionary parameters

Parameter	Description
Use Unrecognized Char	Select this check box ...
Correction Mode Type	• Safe: • Normal: • Best match: • Custom:
Ignore Case	Select this check box ...
Exact Word	Select this check box ...
Out Mode Type	• No change: • Normal: • Always reject: • Custom:
Source	Click ⋯ to select a file ...

Engine parameters

Use these properties to specify how the Equalizer algorithm will work with the engine results.

Property	Description
Confidence for first guess	If you do not wish to rely on Equalizer's conversion statistics for engine results, you can provide your own conversion scale.
	For the first guess of all engine results, provide a string to convert confidence levels to Equalizer rankings (0-3). Separate confidence levels with commas. The string you provide is a series of confidence percentages, which correspond to the Equalizer scores 0, 1, 2, 3, in this order.

Chapter 16

Test virtual engines

After you have set up new OCR engines and virtual engines, you can use the engine tester to generate statistics regarding engine to evaluate the results. Testing allows you to:

- Check how well an engine or virtual engine recognizes characters in a typical form.
- Open an image and apply a virtual engine to a certain region.
- Ascertain whether the engine or virtual engine mistakenly assigns high confidence to badly recognized characters (false positives).
- Ascertain how well a certain engine or virtual engine works on different forms and different areas of a form
- Compare the results from different engines or virtual engines on the same form.

Use the testing results to locate segmentation and recognition problems to improve how engines handle information. Then, using the results that you have generated with the engine tester, change relevant properties, attributes or enhancement methods until results improve.

Test an engine

1. Click to select the engine or virtual engine.
2. On the OCR Working Area toolbar, click **Open image** .
3. In the **Open** dialog box, click on the file name to select the image file and click **Open**.
4. The image file appears in the Working Area, and the engine testing toolbar is activated.
5. Click **Recognition Test** to open the **Engines Tester** dialog box.

The toolbar contains the following buttons:

Button	Description
	Open an image batch file for testing.
	Open the **Engines Tester** dialog box.
	Use these buttons to page through the form. Click the left most button to return to the first page. Click the right most button to jump to the last page in the form. Use the two middle buttons to move one page backwards or forwards, respectively. The page location in the form is displayed in the central display box.
	Resize the image: zoom out, return to the original size, or zoom in.

Define test regions

After you open the **Engines Tester** dialog box, you must define a test region, where recognition will be performed. The test region is the specific region that you select in the form that will undergo the various types of data processing.

1. In the Working Area, click anywhere in the image.

   The pointer becomes a cross.

2. Drag the cross to create a rectangle over the desired region, until you reach the size that you want.

   The area is highlighted in yellow.

3. Release the mouse button.

To deselect the region, click anywhere outside the region.

> ⓘ Make sure to choose a bar code to test a bar code engine setup, numbers to test numerical recognition, and so on.
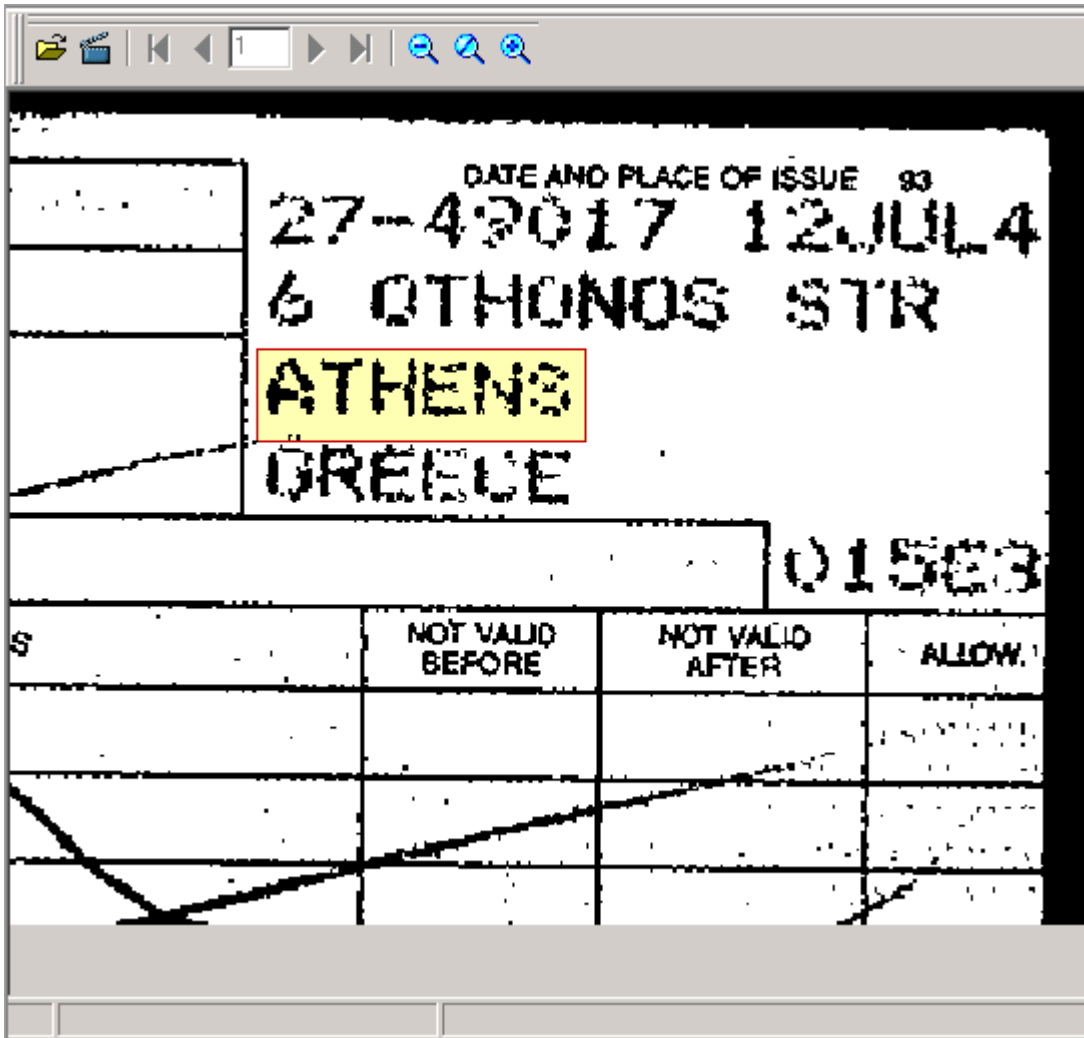
Run tests

The test region you selected in the OCR Working Area is displayed at the top of the **Engines Tester** dialog box.
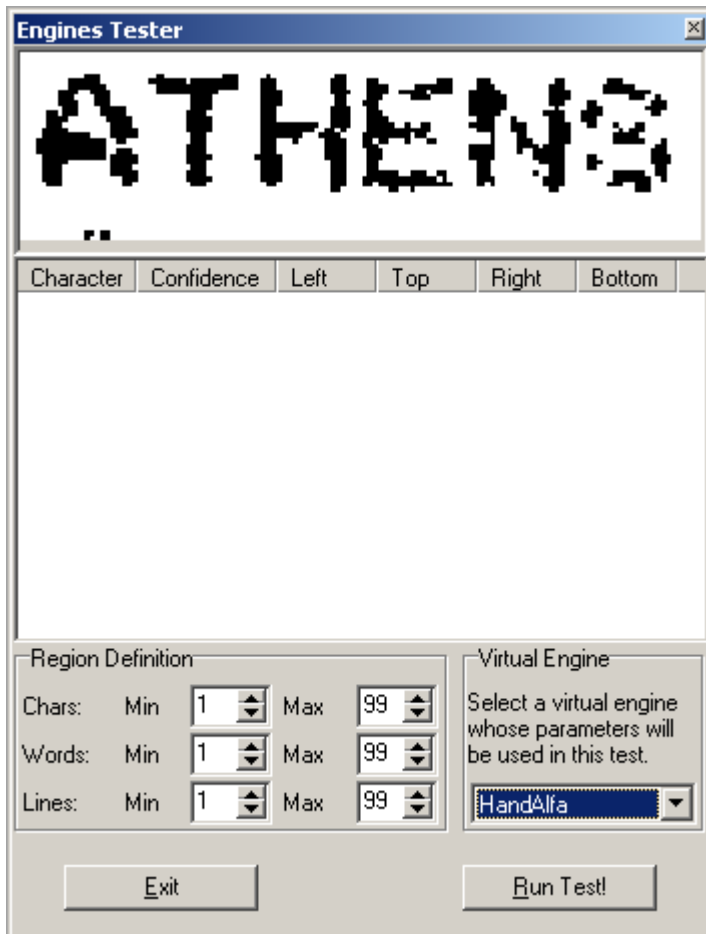
You must now set test parameters, then run the test.

Property	Description
Region Definition	Specify a minimum and maximum number of characters, words and lines expected for the field being tested.
Virtual Engine	If you are not testing a virtual engine (consisting of several recognition engines working together), this drop-down list appears. You must select a test engine using the drop-down list.
	When testing a virtual engine, this drop-down list does not appear.

When you have chosen a test region and specified field definitions and virtual engine, click **Run Test** at the bottom of the **Engines Tester** dialog box.

Engine test results

The results are displayed in the central area of the **Engines Tester** dialog box. This area provides cumulative engine result states for individual fields. Result states are displayed in the following columns:

- **Character**: The character recognized.
- **Confidence**: Rating of confidence that the character was correctly identified by OCR voting.
- **Left**, **Top**, **Right**, **Bottom**: Coordinates of the character recognized.
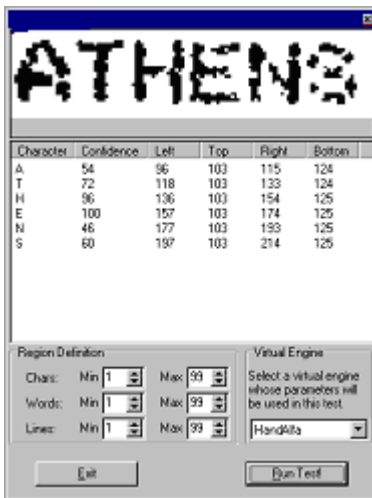
If the results are not satisfactory, change the engine, the region definitions and/or virtual engine parameters, and retest until you receive satisfactory results.

Workflow design

You can define the flows of the application in the Workflow Designer. The Workflow Designer is a graphic interface that allows you to define a dynamic ordering of eFlow modules and all relevant parameters for each module. There may be one or more flows, but all the flows are defined here at the same time.

The Workflow Designer allows you great flexibility in workflow setup. With the Workflow Designer, you can:

- Choose the order in which you perform the standard input, processing, validation, exception and export modules
- Clone existing modules
- Add custom modules (new user-defined modules)
- Save a workflow template

Before you can use the Workflow Designer, you must understand your requirements and plan the project. You can then define a flow that matches your project plan. Before setting up a workflow, you should specify create and define set all the necessary objects, properties, and exceptions for the application.

To open the Workflow Designer, on the Design toolbar, click **Workflow Designer** .

Workflow Designer desktop

The Working Area of the Workflow Designer shows a schematic diagram of the flow. Each box represents a workflow module, that is, a station. Changing this diagram changes the flow of stations in the system.
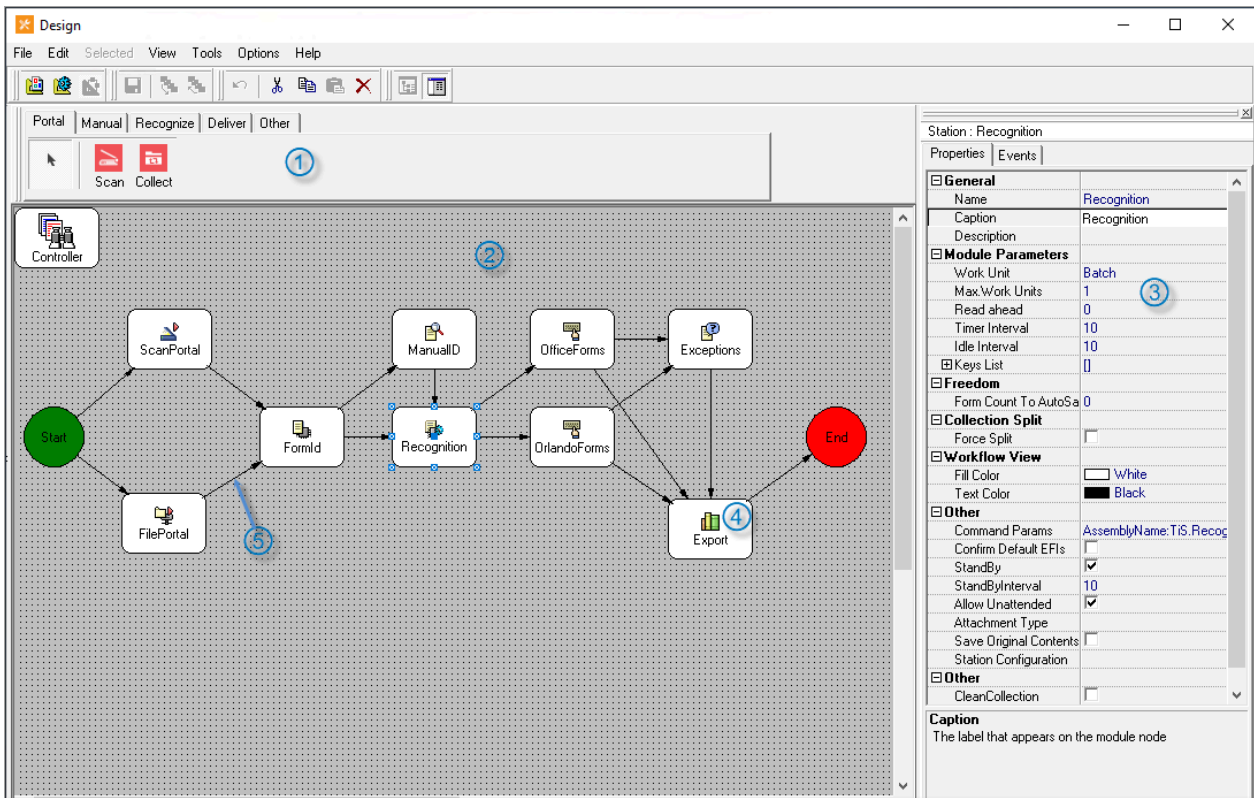
1	Module selection toolbar
2	Working area
3	Object inspector
4	Workflow module
5	Data router

The Workflow Designer profoundly affects how the runtime stations operate. It defines:

- The stations are used
- The number of stations are used (for example, one or two Validate stations)
- The order in which the stations operate

There are several predefined flow templates. You can choose the one that most resembles your project plan, and then adjust it by adding or removing stations and changing the routes between them. To use a template, on the **File** menu, select **Workflow Template**, then select a template.

When you open the Workflow Designer, an empty workflow is displayed. The **Start** object marks the beginning of the workflow, and the End object marks the end of the workflow. You must now add workflow modules (stations) and data routers, which define the way in which flows and forms pass between modules.
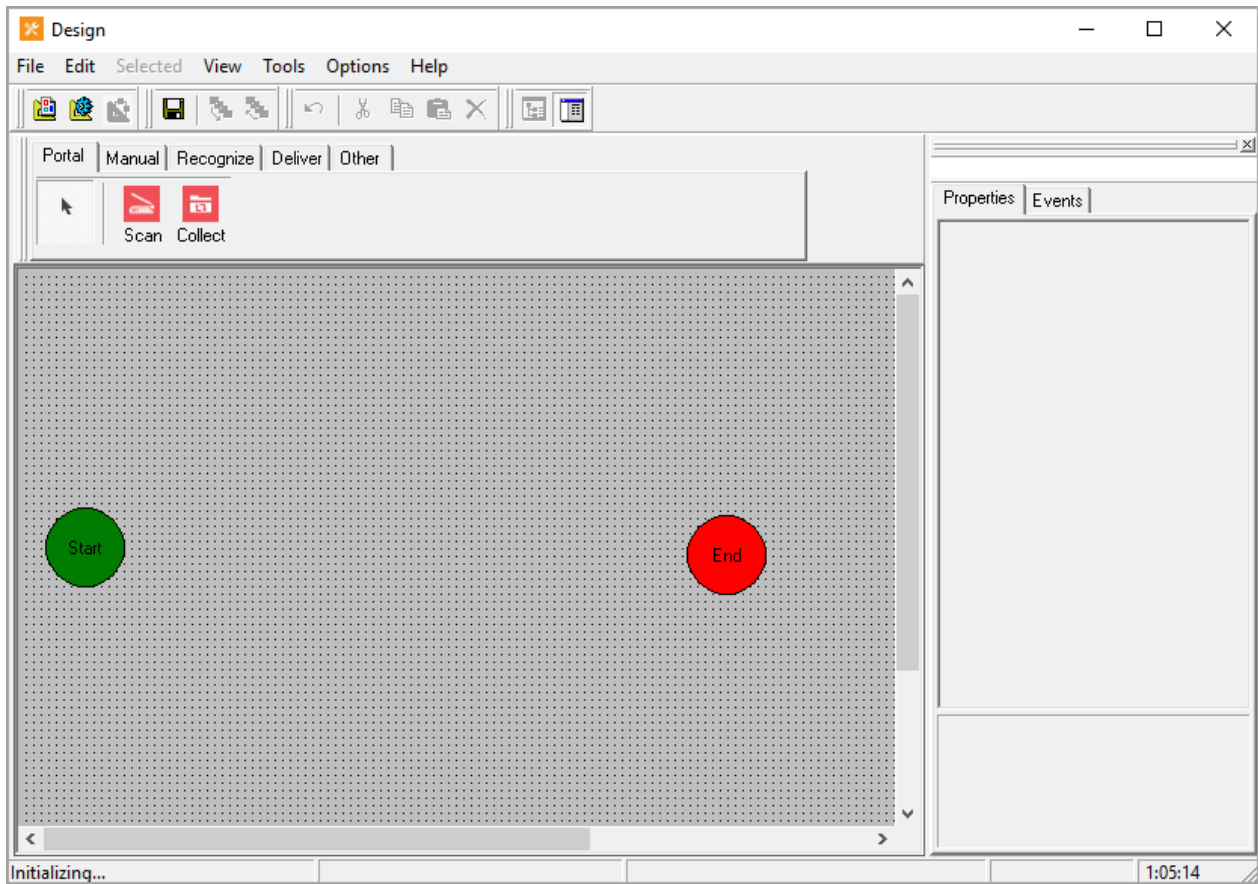
Add modules to a workflow

1. On the module selection toolbar, click on the module.
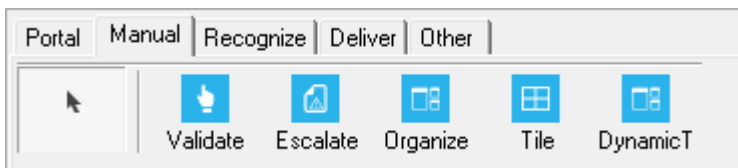
   See Workflow modules for information on the available modules.

2. In the Working Area, click where you want to place the module.

   The icon for the module appears in the Working Area with a default name. The Object Inspector displays the properties and events.

3. Set the module properties. See Workflow module properties for more information.

4. To move a module, click on the module to select it, then drag and drop it to the desired location.

5. To copy, paste, cut or delete a module:

    a. Click on the module to select it.

    b. Right-click the highlighted module to open the context menu and select **Copy**, **Paste**, **Cut** or **Delete**.

Define data routers (links between modules)

You must create links between modules to define the way in which flows and forms will pass between them. These links are called data routers. Every module fetches (gets) collections from a data router at the start of its operation and returns (puts) collections back to a data router at the end of its operation.

You can filter which collections and forms pass from module to module using routing and exception rules.

Add a data router

1. Click on the first module to select it.
    A blue square appears in the centre of the module.
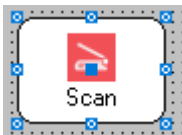
2. Position the mouse pointer over the blue square. The pointer becomes a cross.

3. Press and hold the left mouse button, then drag the pointer to the second module and release the mouse button.
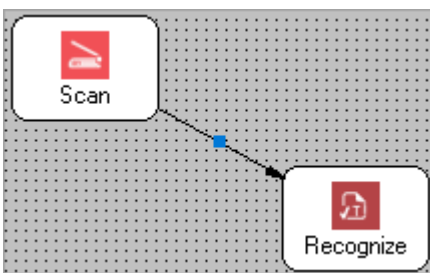
4. In the Object Inspector, define the data router properties.

5. Repeat these steps to connect all modules with data routers. Do not forget to also add data routers to the **Start** and **End** objects.

Data router properties

Property	Description
Name	The name of the data router. All names should be unique. The **Name** field is case sensitive.
Description	A short description of the router that explains its purpose.
RoutingRules	Click ⋯ to the right of this property to define routing and exception rules.

Routing and exception rules

You can use routing rules to clearly specify down to the field level which data can be routed along the data router. For example, you can specify that certain fields will not be passed along to the next station, for security or other reasons.

> 🛑 Great care must be taken in setting this property to ensure smooth workflow results. The proper functioning of the workflow depends on the source station and the target station. You must consider this carefully when adding a data router.

1. Click on the data router.

2. Right-click the data router to open the context menu and select **Routing Rules**.
   The **Routing Rules** dialog box opens.
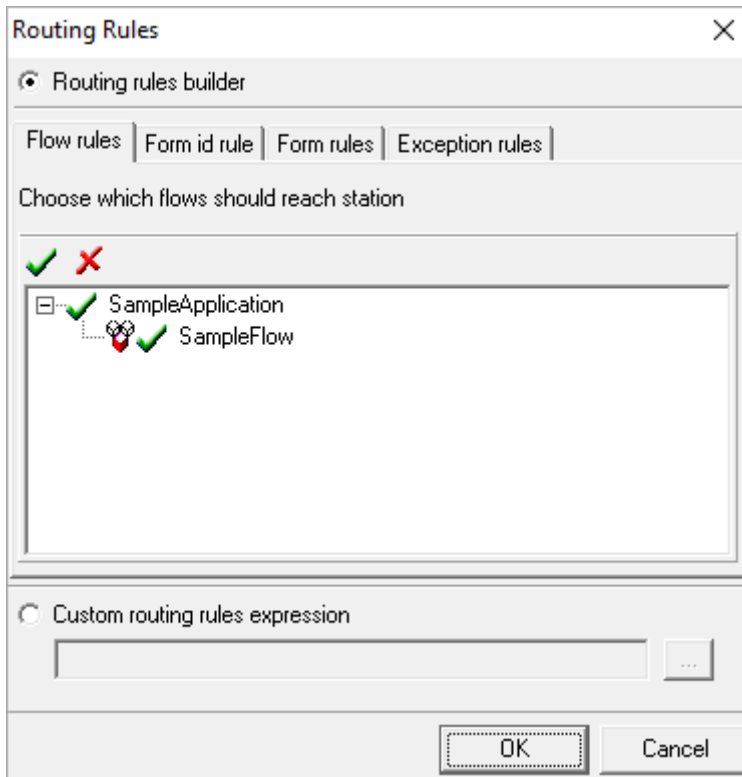
Flow rules

Flow rules determine in which of the application's flows stations participate. The dialog box shows a tree of the active flows. Click the expand ⊞ sign to the left of a flow to display the objects underneath it in the tree.

By default, all flows are selected for routing, indicated by the **Including in routing** ✔ icon to the left of the object.

To exclude a specific flow, select the flow and click **Exclude from the routing rule** ✘ on the **Routing Rules** toolbar.

The data router will be filtered for fields, forms, pages or other disabled objects. Only enabled objects will be allowed to pass through to the next module.

FormID rule

In this tab, you can specify to which forms the routing rules apply:
- **All forms**
- **Recognized forms only**: Forms that are matched to an EFI.
- **Unrecognized forms only**: Forms that are not matched to an EFI.

Form rules
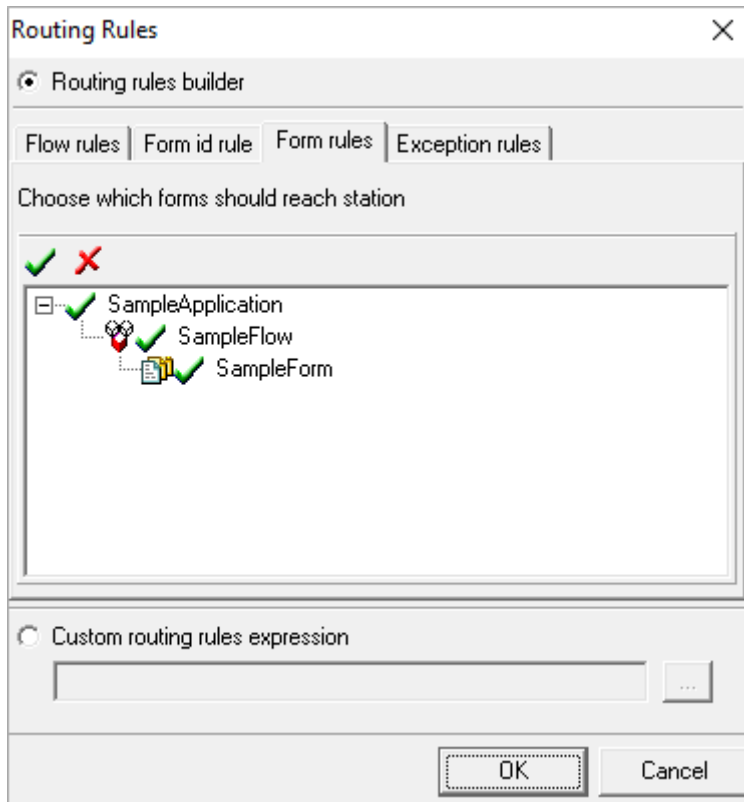
This tab is only enabled when the work unit of the station is defined as the form (instead of the batch).

To specify which forms should be routed along the data router, use **Include** ✔ and **Exclude** ✖ on the **Routing Rules** toolbar.

Click **OK** to complete the selection.

Exception rules
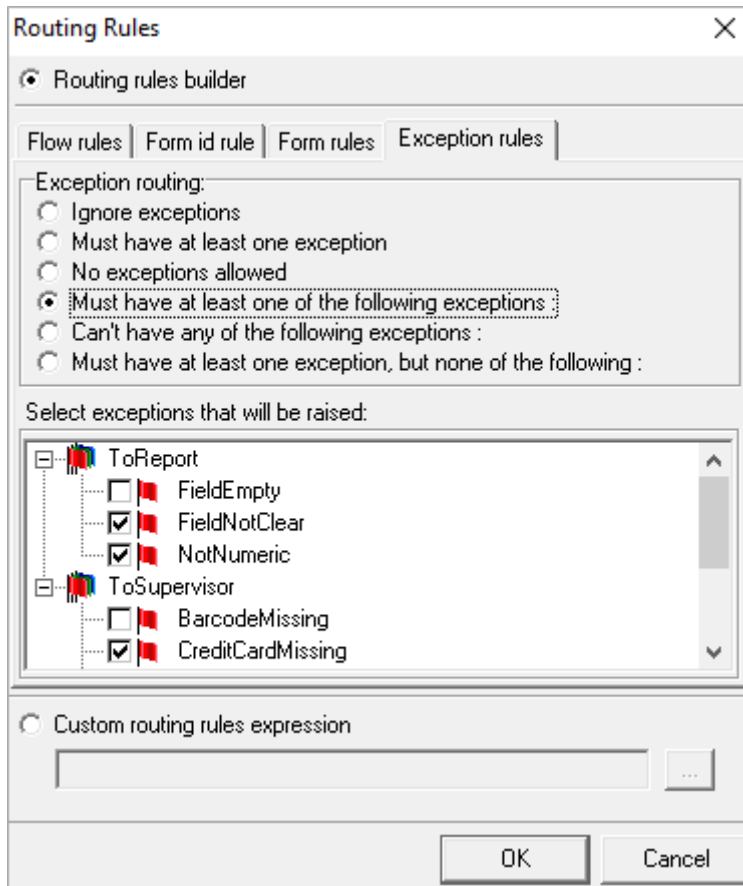
Specify how the data router filters exceptions. The data router will be monitored for collections with matching exceptions. Only matched collections will be allowed to pass through to the next module. See Define exceptions for more information on exceptions.

- **Ignore exceptions**: Select this option if exceptions are irrelevant.
- **Must have at least one exception**: For example, the Escalate station.
- **No exceptions allowed**: For example, the Deliver station.
- **Must have at least one of the following exceptions**: Specify the exceptions by selecting or clearing the check boxes next to the listed exceptions. For example, if there is more than one Escalate station, you can specify which exceptions are routed to which Escalate station.
- **Can't have any of the following exceptions**: Specify the exceptions by selecting or clearing the check boxes next to the listed exceptions.
- **Must have at least one exception, but none of the following**: Specify the exceptions by selecting or clearing the check boxes next to the listed exceptions.

Custom routing rules expression

To create custom rules, select **Custom routing rules expression** at the bottom of the dialog box. Click ⋯ to open the **Rules Editor** dialog box. See Use the Rules Editor for more information.

For example, if a field value is greater than 100, you may wish to send the batch to a certain station. If the same field value is less than 100, you may wish to send the batch to a different station.

Workflow modules

Standard modules

Name	Icon	Description
Scan		The Scan module supplies scanned forms to the system.
Collect		The Collect module supplies image files of previously scanned forms to the system.
Recognize		The Recognize module receives data from the data collection modules for form identification. It automatically identifies a form using empty form images supplied during the design stage. This module is one of the most important parts of eFlow. It is responsible for recognizing as much data as possible. The images undergo image enhancement, automatic form recognition, form removal, and optical character recognition.
Validate		The Validate module receives data from the Recognize module. The exact data it receives depends on the design definitions, but typically it is invalid or unrecognized data. Unrecognized data is completed manually, with or without the help of lookup tables.
Escalate		The Escalate module collects the exceptions that were marked for batches in the Validate station and presents them to the exception handler for clarification and resolution. The exception handler will resolve the exceptions, clarifying information that was missing or incorrect on the forms.
Deliver		The Deliver module exports data to XML and ASCII formats, and additionally exports both B&W and color images. During runtime, the Deliver module receives the data, converts it to the defined format, and transfers it to the export directory.

Additional modules

Name	Icon	Description
Organize		The Organize module allows operators to delete pages from a multi-page batch, insert new pages into a new multi-page batch (also by scanning), and resort or rescan pages.
Tile		Displays in tile format all data recognized by character (for example, all characters recognized as an A). Allows you to identify which characters are correct and which are not.
Custom		User-defined module.

Workflow module properties
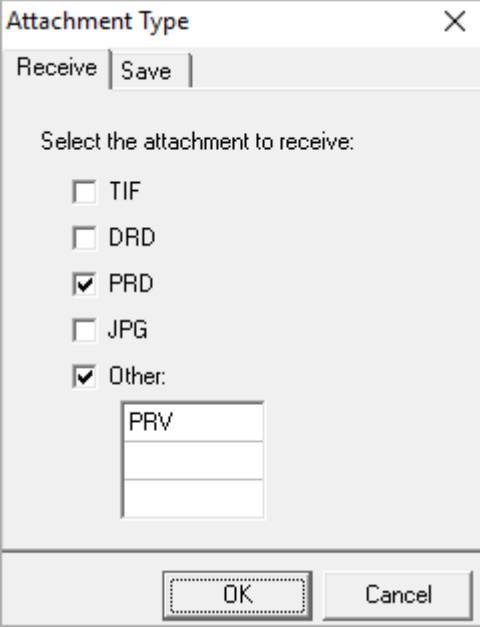
Every module has its own properties, and many have particular events associated with the specific module. Specify these properties to tailor the module activity to your needs.

You define a module's properties in the Object Inspector. When you select a module by clicking on it in the Workflow Designer working area, its properties are automatically displayed.

General module properties

The general module properties listed below are available for every module.

Property	Sub-property	Description
General	Name	The name of the module. The name should be alphanumeric. This name appears automatically in the **Description** box in the Object Inspector after you create a module. You can change the description if you want to add additional information that may be helpful to you.
	Description	A short description of the module.
	Caption	The name of the module that appears on the screen throughout the application.
Module Parameters	Max Work Units	Specify the maximum number of collections that the module receives upon request.
	Read ahead	Improves performance by performing the Get function in advance for the number of work units specified here.
	Timer Interval	The timer interval in seconds. Related to the OnTimer event, which is fired every time the value of the timer interval second counter is reached when a station is idle.
	Idle Interval	The idle interval in seconds. Related to the OnIdle event, which is fired every time the value of the idle interval second counter is reached when a station is idle.
	Keys List	Specify a list of function keys that will trigger the OnKey events per station.
Freedom	Form Count To AutoSave	Holds the information from a defined number of forms in the Learning station before being transferred to the learning database. For example, if this value is set to 3, the Learning station will keep the learned information from 3 forms before storing the information in the database. This function is useful when there are large files that take a long time to save to the database.
Workflow View	Fill Color	Fill the module box with the specified color.
	Text Color	Display the module box text in the specified color.
Other	Command Params	Additional parameters defining module behavior during runtime.
	Confirm Default EFIs	Select this check box to use the default EFI if a form is not recognized.

Property	Sub-property	Description
	Attachment Type	Define which file types are received or saved by this module.
	Save Original Contents	Save the contents before export.

Scan module

In addition to the general module properties, the Scan module has the following properties:

Property	Description
Enable WEB	Allow the station to run in a web browser.
Station Configuration	Click […] to open a dialog box in which you can set station-specific parameters.

Collect module

In addition to the general module properties, the Collect module has the following properties:

Property	Description
StandBy	Automatic stations are usually set to standby in production so that incoming forms are processed automatically by the station.
StandBy Interval	Define how often stations on standby should look for new batches to process.
Allow Unattended	Select this check box to allow the station to run automatically.

Property	Description
Station Configuration	Click ... to open a dialog box in which you can set station-specific parameters.

Recognize module

In addition to the general module properties, the Recognize module has the following properties:

Property	Description
Work Unit	Specify whether the module works with collections or individual forms.
Read Ahead	Improves performance by performing the Get function in advance for the number of work units specified here.
StandBy	Automatic stations are usually set to standby in production so that incoming forms are processed automatically by the station.
StandBy Interval	Define how often stations on standby should look for new batches to process.
Allow Unattended	Select this check box to allow the station to run automatically.
Station Configuration	Click ... to open a dialog box in which you can set station-specific parameters.

Validate module

In addition to the general module properties, the Validate module has the following properties:

Property	Description
Work Unit	Specify whether the module works with collections or individual forms.
Read Ahead	Improves performance by performing the Get function in advance for the number of work units specified here.
Template Teaching Mode	Define whether learning may be performed in the station. • Completion with learning: The Learning module is available to typists. • Completion only: The Learning module is not available.
Enable WEB	Allow the station to run in a web browser.
Station Configuration	Click ... to open a dialog box in which you can set station-specific parameters.

Escalate module

In addition to the general module properties, the Escalate module has the following properties:

Property	Description
Work Unit	Specify whether the module works with collections or individual forms.
Read Ahead	Improves performance by performing the Get function in advance for the number of work units specified here.

Property	Description
Enable WEB	Allow the station to run in a web browser.
Station Configuration	Click ⊡ to open a dialog box in which you can set station-specific parameters.

Deliver module

In addition to the general module properties, the Deliver module has the following properties:

Property	Description
Read Ahead	Improves performance by performing the Get function in advance for the number of work units specified here.
StandBy	Automatic stations are usually set to standby in production so that incoming forms are processed automatically by the station.
StandBy Interval	Define how often stations on standby should look for new batches to process.
Allow Unattended	Select this check box to allow the station to run automatically.
Working Offline	If selected, the module will work offline.
Export Form Condition	Click ⊡ to specify whether exceptions will prevent forms from being exported. • **Export regardless of any exceptions**: Forms will always be exported even if there are exceptions. • **Any exception will prevent form export**: Any exception will prevent forms from being exported. • **The following exceptions will prevent form export**: Specify the exceptions by checking or clearing the check boxes next to the listed exceptions.
Station Configuration	Click ⊡ to open a dialog box in which you can set station-specific parameters.

Organize module

In addition to the general module properties, the Organize module has the following properties:

Property	Description
Read Ahead	Improves performance by performing the **Get** function in advance for the number of work units specified here.
Station Configuration	Click ⊡ to open a dialog box in which you can set station-specific parameters.

Tile module

In addition to the general module properties, the Tile module has the following properties:

Property	Description
Work Unit	Specify whether the module works with collections or individual forms.

Property	Description
Read Ahead	Improves performance by performing the Get function in advance for the number of work units specified here.
Enable WEB	Allow the station to run in a web browser.
Station Configuration	Click ⬚ to open a dialog box in which you can set station-specific parameters.

Run module

In addition to the general module properties, the Run module has the following properties:

Property	Description
Work Unit	Specify whether the module works with collections or individual forms.
Read Ahead	Improves performance by performing the **Get** function in advance for the number of work units specified here.
StandBy	Automatic stations are usually set to standby in production so that incoming forms are processed automatically by the station.
StandBy Interval	Define how often stations on standby should look for new batches to process.
Allow Unattended	Select this check box to allow the station to run automatically.

Custom module

In addition to the general module properties, the Custom module has the following properties:

Property	Description
Work Unit	Specify whether the module works with collections or individual forms.
Read Ahead	Improves performance by performing the **Get** function in advance for the number of work units specified here.
Executable	The complete path of the module's executable file.
StandBy	Automatic stations are usually set to standby in production so that incoming forms are processed automatically by the station.
StandBy Interval	Define how often stations on standby should look for new batches to process.
Allow Unattended	Select this check box to allow the station to run automatically.

View a workflow in the Control module

The modules selected and put together as a workflow in the Workflow Designer window are shown in the Control module and run using eFlow Launch/Launch Pro. Before running the workflow, you must save the workflow and exit the Design module. When you open the Control module, all the modules from the workflow will be available in the Control window.

Event functions

You can define your own action routines, called events, that run on object contents. These events are executed when the operator performs an action in the Validate station or other stations, such as entering or exiting a field, or pressing a specific function key. The available function keys are: F1-F12 and combinations of Ctrl, Alt, and Shift. Events are defined for the station interfaces.

This section contains a list of the system events that can be programmed using the .NET managed functions.

Build an event

You can create events either by writing Win32 DLLs, or by using one of the two custom .NET environments:

- VS IDE environment for VB.Net projects
- #SharpDev IDE environment for C# projects

After you have coded the event, assign it to the appropriate object or workflow module in the Object Inspector:

1. In the Application or Workflow Designer, click the object for which you wish to create an event.

2. Select **Events** tab in the Object Inspector.

3. In the event field, perform one of the following steps to add your code:
    - Enter the full method name:
        - for Win 32: <DLL_name>:<Function_name>
        - for .Net: <Assembly_name>:<Class_name>:<Method_name>
    - Click ⬚ and select an assembly file.

Field events

(*) The first parameter for each event, before the parameters below, is CSM (ITisClientServicesModule type).

Name	Description	Parameter	Parameter Type	Parameter Description
OnChange	Triggered when the contents of the field are changed.	[in] FieldData	ITisFieldData	Dynamic tree node (after the change).
		[in] FieldValue	String	The original field value.
OnEnter	Triggered when the field is entered.	[in] FieldData	ITisFieldData	Dynamic tree node.
		[in] FieldValue	String	The field value.
OnExit	Triggered when the field is exited.	[in] FieldData	ITisFieldData	Dynamic tree node.
		[in] FieldValue	String	The field value.
OnExiting	(Validate only) Triggered when the field is exited.	[in] FieldData	ITisFieldData	Dynamic tree node.
		[in - out] bCancel	Boolean	Whether to cancel exiting from the field.
OnKey	Triggered when a key is pressed.	[in] FieldData	ITisFieldData	Dynamic tree node (before the change).
		[in] FieldValue	String	The field value (before the change).
		[in - out] Key	Short	The code of the key pressed.
		[in - out] Modifiers	Short	Shift state.

ℹ The same (corresponding) event prototypes are used for the OnAnyFieldEnter, OnAnyFieldChange, OnAnyFieldExit and OnAnyFieldKey assigned to the form.

Field validation event

Name	Description	Parameter	Parameter Type	Parameter Description
OnValidate		[in] FieldData	ITisFieldData	Dynamic tree node.
		[in] FieldValue	String	The original field value.
		Return value	Long	0 = valid Other = invalid field

Field group events

Name	Description	Parameter	Parameter Type	Parameter Description
OnEnter	Triggered when the field group is entered.	[in] FieldGroupData	ITisFieldGroupData	Dynamic tree node.
OnExit	Triggered when the field group is exited.	[in] FieldGroupData	ITisFieldGroupData	Dynamic tree node.

Runtime module common events

All the runtime modules have the following events:

Name	Description	Parameter	Parameter Type	Parameter Description
OnIdle	Invoked whenever the idle time of any station has reached the time limit specified in the Idle Interval property.	<none>		
OnLogin	Performed before the user login session starts.	<none>		
OnLogout	Performed when the module is closed.	<none>		
OnTimer	Invoked whenever the idle time of any station has reached the time limit specified in the Timer Interval property.	<none>		
OnFunctionKey	Performed when a defined function key (F1-F12) is pressed.	[in] Key	short	
		[in] Modifiers	short	Shift state.
OnPrePut Collections	Performed just before the module returns the forms/batches to the system.	[in-out] CanPut	boolean	Allow the action to occur.

Name	Description	Parameter	Parameter Type	Parameter Description
OnPostPut Collections	Performed after the module returns the forms/batches to the system.	<none>		
OnPreDiscard Collections	Invoked before the module discards one or more collections.	[in-out] CanDiscard	Boolean	Allow the action to occur.
OnPostDiscard Collections	Invoked when the module discards one or more collections.	<none>		
OnPreSave Collections	Performed when a form is about to be saved into the database.	[in-out] CanSave	boolean	Allow the action to occur.
OnPostSave Collections	Performed after a collection is saved into the database.	<none>		
OnPreUndo Collections	Performed when the module requests to undo the creation of a collection just before the collection is undone. Used by input stations only.	[in-out] CanUndo	Boolean	Allow the action to occur.
OnPostUndo Collections	Performed after the creation of a collection is undone. Used by input stations only.	<none>		
OnPreCreate Collections	Performed just before the collection is created. Used by input stations only.	[in-out] CanCreate	boolean	Allow the action to occur.
OnPostCreate Collections	Performed after the collection is created. Used by input stations only.	<none>		
OnPreDelete Collections	Performed just before the collection is deleted. Used by export stations only.	[in-out] CanDelete	boolean	Allow the action to occur.
OnPostDelete Collections	Performed after the collection is deleted. Used by export stations only.	<none>		

Runtime module events

Collect module

In addition to the runtime module common events, the Collect module has the following events:

Name	Description	Parameter	Parameter Type	Parameter Description
OnFlowSelect	Performed when the flow to be used is selected or modified.	[in] FlowName	String	
OnStartManual FileGather	Performed when the Run button is clicked and manual gathering of files starts.	[in] LinkedFileNames	Safe Array	Always an empty array (used only by efAutoGate).
		[in] NonLinkedFileNames	Safe Array	Contains strings.
OnStartAutoFile Gather	Performed when the polling timer is reached, and automatic file polling starts.	[in] RemoteLocationType	short	0 - LAN 1 - FTP
		[in-out] searchPath	string	
		[in-out] AdditionalData	variant	
OnEndFileGather	Performed after file gathering has ended.			
OnPreFileRead	Performed before file processing occurs (from the local temporary files).	[in-] OriginalExtFile Name	String	The original (remote) extracted file name.
		[in-out] FileName	String	Extracted file names. (File names are relative to the search path).
OnPostFileRead	Performed after file processing occurs.	[in] FileName	String	(File names are relative to the search path).
		[in] ReadStatus	Long	

Name	Description	Parameter	Parameter Type	Parameter Description
OnFillInputForm	Called before a new batch is inserted into the system, to enable automatic input form filling.			N/A in Ver. 1.0.0.15.
OnPrePageHandle	Performed before a page is read.	[in] PageNumber	Short	1 based.
		[in] FileName	String	
		[in-out] ExternalPageTest	Boolean	A flag indicating if the user is in charge of checking if the page is a separator page. When set to True, the page type is not checked by the application, and should be checked by OnPostPage.
		[in-out] AbortSession	Boolean	Terminates the gathering of the image files and does not delete the original image files.
OnPostPageHandle	Performed after a page is read and is resident in memory.	[in] PageNumber	Short	
		[in] FileName	String	
		[in] hBitmap	HBITMAP	
		[in-out] PageType	enum	Separator/Blank/Normal.
		[in-out] DiscardPage	Boolean	
		[in-out] AbortSession	Boolean	

For more information on the Collect module, see Collect module.

Scan module

In addition to the runtime module common events, the Scan module has the following events:

Name	Description	Parameter	Parameter Type	Parameter Description
OnFlowSelect	Performed when a flow is selected.	[in-out] FlowName	String	

Name	Description	Parameter	Parameter Type	Parameter Description
OnPreInputForm	Performed before the input form is displayed. Used to set the input form's values, or to display customized forms instead of built-in input forms.	[in-out] ShowForm	Boolean	
OnPostInputForm	Performed after the input form is displayed. Used to cancel the input operation based on the appropriateness of the input form's values.	[in-out] CanClose	Boolean	Allow the action to occur.
OnScanSession Reset	Performed when the scanning button is pressed.	[in-out] CanReset	Boolean	Allow the action to occur.
OnScanSession Continue	Performed when the **Continue** button is pressed (while in pause mode).	[in] LastBatchNumber	integer	Relative to session. 1 based.
		[in] LastFormNumber	integer	Relative to batch. 1 based.
		[in] LastPageNumber	integer	Relative to form. 1 based.
		[in-out] CanContinue	Boolean	If False, the session remains paused.
OnScanSession Stop	Performed when the **Stop scan session** button is pressed.	[in-out] CanStop	Boolean	If False, the session does not stop.
OnScanSession Done	Performed when the batch scan is terminated normally.			
OnScanSession Error	Performed when the batch scan is terminated abnormally (due to an error).	[in] StopReason	long	Relative to session. 1 based.
		[in-out] AbortSession	Boolean	
OnPrePageScan	Performed before a page is scanned.	[in] PageNumber	short	Relative to session. 1 based.

Name	Description	Parameter	Parameter Type	Parameter Description
		[in-out] ExternalPageTest	Boolean	A flag indicating if the user is in charge of checking if the page is a separator page. When set to True, the page type is not checked by the application, and should be checked by OnPostPage.
		[in-out] PrepareBitmap	boolean	
		[in-out] AbortSession	Boolean	
OnPostPageScan	Performed after a page is scanned and is resident in memory.	[in] BatchNumber	short	Relative to session. 1 based.
		[in] FormNumber	short	Relative to batch. 1 based.
		[in] PageNumber	short	Relative to form. 1 based.
		[in] FileName	String	The local image file name, excluding the last page (not added yet). If both B&W and color images are scanned, this will be the B&W file name. The color image file name in such a case can be derived from the same file name after the extension is replaced.
		[in] hBitmap	HANDLE	HBITMAP
		[in-out] PageNumber	short	
		[in-out] PageType	enum	Separator/Blank/ Normal.
		[in-out] DiscardPage	Boolean	
		[in-out] AbortSession	Boolean	

Name	Description	Parameter	Parameter Type	Parameter Description
OnPreRescanPage	Performed before a page is re-scanned.	[in] BatchNumber	short	Relative to session. 1 based.
		[in] FormNumber	short	Relative to batch. 1 based.
		[in] PageNumber	short	Relative to form. 1 based.
		[in-out] CanRescan	Boolean	
OnDeleteScanned Batch	Performed when the **Delete Batch** menu option is selected.	[in] BatchNumber	short	1 based.
		[in-out] CanDelete	Boolean	
OnDeleteScanned Form	Performed when the **Delete Form** menu option is selected.	[in] BatchIndex	Short	Relative to session. 1 based.
		[in] FormIndex	Short	Relative to batch. 1 based.
		CanDelete	Boolean	
OnDeleteScanned Page	Performed when the **Delete Page** menu option is selected.	[in] BatchNumber	short	Relative to session. 1 based.
		[in] FormNumber	short	Relative to batch. 1 based.
		[in] PageNumber	short	Relative to form. 1 based.
		[in-out] CanDelete	Boolean	

For more information on the Scan module, see Scan module.

Validate module

In addition to the runtime module common events, the Validate module has the following events:

Name	Description	Parameter	Parameter Type	Parameter Description
OnExceptionListDisplay	Performed when the **Exceptions List** dialog box is displayed. Performed when the exceptions list display is canceled.	[in] FieldData	ITisFieldData	Dynamic tree node.
		[in-out] ShowDefScreen	Boolean	Show default screen.

Name	Description	Parameter	Parameter Type	Parameter Description
OnExceptionListCanceled	Invoked when an exception is set.	[in] FieldData	ITisFieldData	Dynamic tree node.
OnExceptionSet	Performed when the **Exceptions List** dialog box is displayed.	[in] FieldData	ITisFieldData	Dynamic tree node.
OnExceptionListDisplay	Performed when the **Exceptions List** dialog box is displayed. Performed when the exceptions list display is canceled.	[in] FieldData	ITisFieldData	Dynamic tree node.

Escalate module

In addition to the runtime module common events, the Escalate module has the following events:

Name	Description	Parameter	Parameter Type	Parameter Description
OnExceptionListDisplay	Performed when the **Exceptions List** dialog box is displayed. Performed when the exceptions list display is canceled.	[in] FieldData	ITisFieldData	Dynamic tree node.
		[in-out] ShowDefScreen	Boolean	Show default screen.
OnExceptionListCanceled	Invoked when an exception is set.	[in] FieldData	ITisFieldData	Dynamic tree node.
OnExceptionSet	Invoked just before user moves to the next exception.	[in] FieldData	ITisFieldData	Dynamic tree node.
OnPreGotoException	Invoked after user moves to the next exception. Performed when the **Exceptions List** dialog box is displayed.	[in] FieldData	ITisFieldData	
		[in-out] CanGo	Boolean	
OnPostGotoException	Performed when the exceptions list display is canceled.	[in] FieldData	ITisFieldData	Dynamic tree node.

Deliver module

In addition to the runtime module common events, the Deliver module has the following events:

Name	Description	Parameter	Parameter Type	Parameter Description
OnPreConvertBatch	Invoked before a batch is converted.	[in-out] CanConvert	Boolean	
OnPostConvertBatch	Invoked after a batch is converted.	[none]		
OnXMLFormCreated	Performed when the XML form is created.	[in] XMLForm	IXMLDOMElement	
OnASCIILineCreated	Performed when the ASCII file is created.	[in-out] LineBuffer	string	
		[in-out] CanExport	Boolean	
OnExceptionLine Created	Performed when the ASCII file is created.	[in-out] LineBuffer	string	
		[in-out] CanExport	Boolean	
OnPreWriteXML	Performed before the XML file is written.	[in] XMLDoc	IXMLDOM Document	
		[in-out] WriteXML	Boolean	
OnPostWriteXML	Performed after the XML file is written.	[none]		
OnPreWriteFile	Triggered before any non-XML file is written to the local drive.	[in] FileType	enum	Exception/ASCII/INI.
		[in-out] Buffer	VARIANT	"PChar*"
		[in-out] WriteFile	Boolean	
OnPostWriteFile	Triggered after any non-XML file is written to the local drive.	[in] FileType	enum	Exception/ASCII/INI.
OnPreBatchLaunched	Triggered before the batch is launched.	[in] CollectionName	string	Required - since this may be on a separate thread. In order to cancel the launch - delete local files.
OnPostBatchExported	Triggered after the batch is launched and is received by the target.	[in] CollectionName	string	Required - since this may be on a separate thread.

For more information on the Deliver module, see Deliver module.

Control module

In addition to the runtime module common events, the Control module has the following events:

Name	Description	Parameter	Parameter Type	Parameter Description
OnPreManage Collections		[in] oAction	ITisManageCollection Action	
		[in-out] bCanDoAction	Boolean	
		[in-out] bDetailsNeeded	Boolean	
OnManage CollectionsDetails		[in] oAction	ITisManageCollection Action	
		[in] oActionInfo	ITisManageCollection ActionInfo	
		[in-out] bCanDoAction	Boolean	
OnPostManage Collections		[in] oAction	ITisManageCollecti onAction	