Perceptive Intelligent Capture for Transcripts

Installation and Setup Guide

Version: 1.3.x

Written by: Perceptive Intelligent Capture
Date: September 2016

perceptivesoftware
from Lexmark

Table of Contents

What is Perceptive Intelligent Capture for Transcripts?

Perceptive Intelligent Capture (PIC) for Transcripts captures information from transcripts fields such as school name, student's name, birthdate, GPA, and grades and stores the information as a transcript document.

The document is then available to use with your institution's student information system and can be shared among different departments. You can also store additional documents with a transcript, such as a letter of recommendation.

Perceptive Intelligent Capture for Transcripts contains document type support for high school and college transcripts, as well as Navy, Marine Corps, Army, and Joint Services transcripts.

What are the supported languages and locations?

Perceptive Intelligent Capture for Transcripts supports transcripts presented in English from United States-based institutions.

What are the PIC for Transcripts components?

PIC for Transcripts includes the following components.

- Perceptive Visibility for auditing and reporting purposes
- Perceptive Verifier for document quality-assurance purposes
- Perceptive Content (optional)

What is the Intelligent Capture for Transcripts solution architecture?

You can implement PIC for Transcripts as an early archiving solution where transcript documents are scanned directly into an existing document management solution (DMS). A copy of the document, one file per transcript, and the URL are passed to the runtime server. For documents requiring client processing, the ID of the client must also be incorporated into the image file name. The document is processed by PIC for Transcripts with the metadata exported to Perceptive Content (optional).

Perceptive Visibility is also included for auditing and reporting purposes. Verifier workstations can be used for document quality assurance. Content is used for escalating of exceptions and other admissions functions prior to the transcript being written to the Student Information System (SIS). Data validation occurs through a live connection to the SIS or alternative data source.

What are the transcript fields?

The transcript fields are included within the Transcript document class and are used by all college, high school, and military transcript document types.

Transcript type field

The transcript type field denotes whether the incoming document is a college, high school, or military transcript. The field result is determined by the system automatically, based on the classification results. The field is presented as read-only within Verifier and can only be changed by reclassification of the document.

Institution ID, Site ID, and Internal Institution ID fields

PIC for Transcripts uses an associative search engine to ascertain the issuing institution. Multiple instances of institution master data, each referred to as an institution partition, are also supported.

By pointing Intelligent Capture to an extract of the client's institution master, whether it resides in a flat file or in a database table, the system analyzes the text of the transcript and then selects the closest matching institution record in a fault-tolerant manner that accounts for spelling differences, OCR errors, abbreviations, and institution details embedded within logos on the transcript.

If the system is not confident that the closest matching institution from the extract is the correct institution, the field is marked invalid and the document is sent to Verifier. A Verifier user then chooses to accept this institution or select an alternative using the institution search facility within Verifier.

The institution ID is a mandatory field for all transcript types.

PIC for Transcripts also supports scenarios where the SIS uses an external institution ID for display to a user, but another institution ID internally. In this scenario, the external institution ID is displayed within Verifier, but the system stores the internal institution ID in the internal institution ID field so that both values are available for export to the downstream systems.

Applicant ID, Applicant Site ID, and Internal Applicant ID fields

PIC for Transcripts uses an associative search engine to ascertain the applicant. Multiple instances of applicant master data, each referred to as an applicant partition, are also supported by the solution.

By pointing Intelligent Capture to an extract of the client's applicant master, whether it resides in a flat file or in a database table, the system analyzes the text of the transcript. It then selects the closest matching institution record in a fault-tolerant manner that accounts for spelling differences, OCR errors, and abbreviations.

If the system is not confident that the returned applicant from the extract is the correct applicant, the field is marked invalid and the document is sent to Verifier. A Verifier user then chooses to accept this applicant or select an alternative using the applicant search facility within Verifier.

A Verifier user can manually enter an applicant ID in the ApplicantID field. If the applicant ID exists in the ApplicantASE pool then the ApplicantASE, Last Name, First Name, Middle Name, and DOB fields are automatically updated with the new information on the Verifier form.

The applicant ID is a mandatory field for all transcript types.

PIC for Transcripts also supports scenarios where the SIS uses an external applicant ID for display to a user, but uses another applicant ID internally. In this scenario, the external applicant ID displays within Verifier, but the system stores the internal applicant ID in the internal applicant ID field so that both values are available for export to the downstream systems.

First Name field

This field is used to capture the first name of the applicant. For college and high school transcripts, the first name is populated when the applicant ID is validated by either the runtime server or a Verifier user. If an invalid reason is selected for the applicant ID, the field is manually entered by a Verifier user, and automatic extraction from the transcript image is not performed.

Note Automated extraction of this field is applied for Army, Navy, Marine, and Joint Services transcripts. If the applicant information does not match the extracted value, the field is presented for verification.

Middle Name field

This field is used to capture the middle name of the applicant. For college and high school transcripts, the middle name is populated when the applicant ID is validated by either the runtime server or a Verifier user. If an invalid reason is selected for the applicant ID, the field is manually entered by a Verifier user, and automatic extraction from the transcript image is not performed.

Note Automated extraction of this field is applied for Army, Navy, Marine, and Joint Services transcripts. If the applicant information does not match the extracted value, the field is presented for verification.

Last Name field

This field is used to capture the last name of the applicant. For college and high school transcripts, the last name is populated when the applicant ID is validated by either the runtime server or a Verifier user. If an invalid reason is selected for the applicant ID, the field is manually entered by a Verifier user, and automatic extraction from the transcript image is not performed.

Note Automated extraction of this field is applied for Army, Navy, Marine, and Joint Services transcripts. If the applicant information does not match the extracted value, the field is presented for verification.

DOB field

This field is used to capture the date of birth of the applicant. If the applicant ID was validated by the runtime server and the DOB is present in the Application ASE pool, the DOB from the Applicant pool is populated in the DOB field.

If the applicant was not validated by the runtime server or the DOB is not present in the applicant pool, PIC for Transcripts attempts to extract the applicant's date of birth. The system automatically converts the date of birth on the document, irrespective of how it is expressed, into the designated Verifier output format. The format can be set to DD/MM/YYYY or MM/DD/YYYY using the configuration settings in the BRWDAT table.

If a date is entered manually in Verifier, then conversion does not take place unless the date entered does not match the Verifier output format.

You can configure the system to invalidate the date of birth if the following is true, where X is configurable.

- It is more than x days in the future.
- It falls more than x days prior to the current date.

Machine and local user settings play no part in the system's internal handling of dates. User input into the date field is not subject to the checks previously listed as long as the date entered is valid for the output format.

SSN field

This field is used to capture the social security number of the applicant. If the applicant ID was validated by the runtime server and the SSN is present in the Application ASE pool, the SSN from the Applicant pool is populated in the SSN field.

If the applicant was not validated by the runtime server or the SSN is not present in the Applicant pool, the PIC for Transcripts attempts to extract the applicant's social security number. Only social security numbers matching the ######### or ###-##-#### format are considered valid formats.

Date Issued field

This field is used to capture the issue date of the transcript. The system automatically converts the date issued on the document, irrespective of how it is expressed, into the designated Verifier output format. You can set the format to DD/MM/YYYY or MM/DD/YYYY using the configuration settings in the BRWDAT table.

If a date is entered manually in Verifier, then conversion does not take place unless the date entered does not match the Verifier output format.

You can configure the system to invalidate the date issued if the following is true, where X is configurable.

- It is more than x days in the future.
- It falls more than x days prior to the current date.

Machine and local user settings play no part in the system's internal handling of dates. User input into the date field is not subject to the checks previously listed as long as the date entered is valid for the output format.

Invalid Reason field

This field contains a list of possible exceptions that could prevent a Verifier user from being able to correct a document in its entirety.

The system default is NONE, but a Verifier user may change this value through the field drop-down list when a particular exception is encountered so that the document can be moved out of the Verifier application.

The following table contains a list of the system-delivered invalid reasons, their corresponding rules, when they are selected, and the effect of selecting them.

Invalid Reason	Usage	Effect	Class
APPLICANT NoT FOUND	This invalid reason is selected if the transcript applicant cannot be found using the applicant search function.	**RULE** SETAPPTOVALID The Applicant ID field and Applicant Info fields are set to blank and valid. Verifiers have the ability to manually enter applicant fields, such as First Name, Middle Name, Last Name, SSN, and DOB. An applicant ID is not exported.	Appears in all classes.
INSTITUTION NoT FOUND	This invalid reason is selected if the institution issuing the transcript cannot be found using the institution search function.	**RULE** SETINSTTOVALID The Institution ID field and Institution Info fields are set to blank and valid. An institution ID is not exported.	Appears in all classes.

Invalid Reason	Usage	Effect	Class
INSTITUTION AND APPLICANT NoT FOUND	This invalid reason is selected if the institution issuing the transcript cannot be found using the institution search function and the transcript applicant cannot be found using the applicant search function.	**RULE** SETINSTANDAPPTOVALID The Institution ID, Institution Info, Applicant ID, and Applicant Info fields are set to blank and valid. Verifiers have the ability to manually enter applicant fields, such as First Name, Middle Name, Last Name, SSN, and DOB. An institution ID or applicant ID is not exported.	Appears in all classes.
CUMULATIVE GPA <> CALCULATED GPA	This invalid reason applies only to College transcripts and is selected if the difference in the GPA extracted from the transcript and the GPA calculated from the Coursework table do not fall within a specified tolerance.	**RULE** SETCUMGPATOVALID The Cumulative GPA field is set to valid.	Appears in all classes. Only applicable to college transcripts.
GPA AND APPLICANT INVALID	This invalid reason applies only to College transcripts and is selected if both of the following conditions apply. The difference in the GPA extracted from the transcript and the GPA calculated from the Coursework table do not fall within a specified tolerance. The transcript applicant cannot be found using the applicant search function.	**RULE** SETGPAANDAPPTOVALID The Applicant ID field and Applicant Info fields are set to blank and valid. The Cumulative GPA field is set to valid. Verifiers have the ability to manually enter applicant fields, such as First Name, Middle Name, Last Name, SSN, and DOB. An applicant ID is not exported.	Appears in all classes. Only applicable to college transcripts.
GPA AND INSTITUTION INVALID	This invalid reason applies only to College transcripts and is selected if both of the following conditions apply. The difference in the GPA extracted from the transcript and the GPA calculated from the Coursework table do not fall within a specified tolerance. The institution issuing the transcript cannot be found using the institution search function.	**RULE** SETGPAANDINSTTOVALID The Institution ID field and Institution Info fields are set to blank and valid. The Cumulative GPA field is set to valid. An institution ID is not exported.	Appears in all classes. Only applicable to college transcripts.

Invalid Reason	Usage	Effect	Class
GPA, INSTITUTION, APPLICANT INVALID	This invalid reason applies only to College transcripts and is selected if all of the following conditions apply. The difference in the GPA extracted from the transcript and the GPA calculated from the Coursework table do not fall within a specified tolerance. The institution issuing the transcript cannot be found using the institution search function. The transcript applicant cannot be found using the applicant search function.	**RULE** SETGPAINSTAPPTOVALID The Institution ID field and Institution Info fields are set to blank and valid. The Cumulative GPA field is set to valid. The Applicant ID field and Applicant Info fields are set to blank and valid. Verifiers have the ability to manually enter applicant fields, such as First Name, Middle Name, Last Name, SSN, and DOB. An institution ID or applicant ID is not exported.	Appears in all classes. Only applicable to college transcripts.

After selecting an Invalid Reason, it takes effect when a user presses Enter while in the InvalidReason field or a corresponding field

Hotkey shortcuts are enabled to set the Institution Not Found (Shift+F2) and Applicant Not Found (Shift+F4) invalid reasons.

You can use the settings in the BRWIVRType table to change the text, rule, and export code associated with an invalid reason, as well as add new invalid reasons based on an existing invalid reason rule. The invalid reason rules available are listed in the previous table.

Invalid reason code

The invalid reason code is the value that the system assigns to a selected invalid reason for the purposes of document export so that a downstream workflow or SIS can act upon that code and behave accordingly.

The code against each invalid reason can be set in the BRWIVRType table.

What are the college fields?

The following college fields are included within the College document class.

Cumulative GPA field

This field is used to capture the cumulative GPA from the transcript.

Graduation Date field

This field is used to capture the latest graduation date from the transcript. The system automatically converts the date of birth on the document, irrespective of how it is expressed, into the designated Verifier output format. The format can be set to DD/MM/YYYY or MM/DD/YYYY using the configuration settings in the BRWDAT table.

If a date is entered manually in Verifier, then conversion does not take place unless the date entered does not match the Verifier output format.

You can configure the system to invalidate the date of birth if the following is true, where X is configurable.

- It is more than x days in the future.
- It falls more than x days prior to the current date.

Machine and local user settings play no part in the system's internal handling of dates. User input into the date field is not subject to the checks previously listed as long as the date entered is valid for the output format.

Degree Earned field

This field is used to capture the highest degree earned that is listed on the transcript. For example, if the transcript denotes that an applicant earned both an Associate's and Bachelor's degree, the Bachelor's degree information is extracted.

Coursework field

PIC for Transcripts attempts to capture the following information at coursework line-item detail from college transcripts.

Coursework field	Description
Course Number	This is the course department and number combined. For example, MTH101 uses MTH for the Course Department (Math) and 101 for the Course Number. You can configure PIC for Transcripts to compare extracted course numbers against a database table of known course numbers, including the issuing institution ID. ● If only one record exists for the extracted course number within the database for the issuing institution, the course number is validated. ● If multiple records exist for the course number within the table but only one of the database course descriptions match the extracted description, the course number is validated. If these validations are unsuccessful, the course number is presented for verification. You can configure PIC for Transcripts to add new course number and description combinations to the database table upon export.
Description	This is the description of the course.
Grade	This is the letter grade achieved for the course. Acceptable grades can be configured in the BRWGRD table.
Earned	This is the credit earned for the course.
Grade Points	This is the grade point earned for the course. Grade points are systematically validated by multiplying the Grade * Earned using both 4 and 11-point grading scales.

What are the high school fields?

The following fields are included within the High School document class.

Cumulative GPA field

This field is used to capture the unweighted cumulative GPA from the transcript.

Graduation Date field

This field is used to capture the graduation date from the transcript. The system automatically converts the date of birth on the document, irrespective of how it is expressed, into the designated Verifier output format. The format can be set to DD/MM/YYYY or MM/DD/YYYY using the configuration settings in the BRWDAT table.

If a date is entered manually in Verifier, then conversion does not take place unless the date entered does not match the Verifier output format.

You can configure the system to invalidate the date of birth if the following is true, where X is configurable.

- It is more than x days in the future.
- It falls more than x days prior to the current date.

Machine and local user settings play no part in the system's internal handling of dates. User input into the date field is not subject to the checks previously listed as long as the date entered is valid for the output format.

Class Rank field

This field is used to capture the unweighted class rank from the transcript.

Class Size field

This field is used to capture the unweighted class size from the transcript.

Class Year field

This field is used to capture the class year from the transcript.

Total Credits Earned field

This field is used to capture the total credits earned from the transcript.

Weighted GPA field

This field is used to capture the weighted GPA from the transcript.

Weighted Class Rank field

This field is used to capture the weighted class rank from the transcript.

What are the military fields?

The military fields are included within the Military document class.

Coursework field

PIC for Transcripts attempts to capture the following information at coursework line item detail from Military transcripts.

Coursework field	Description
ACE Identifier	This is the American Council on Education (ACE) identification number.
Course ID	This is the military course ID for the coursework line.
Date Completed	This is the completion date for the coursework line.
Course Title	This is the title of the course for the coursework line.

What are the PIC for Transcripts solution features?

The following features are available within the PIC for Transcripts solution.

- Data export options
- Document management system (DMS) integration
- Solution reporting

What are the data export options?

PIC for Transcripts provides the following standard export options.

- Output of XML files
- Output of TIFF files
- Output of fully text-searchable PDF files

What is document management system (DMS) integration option

PIC for Transcripts supports integration to DMS in both the early and late archiving scenarios.

Early archiving integration

Early archiving means that an image has already been archived prior to reaching Intelligent Capture. In this scenario, Intelligent Capture requires a copy of the archived image with the unique archive document ID embedded into the document filename.

Configuration options in the IMP section in the INI file define whether this unique archive document ID constitutes the entire filename or an underscore-separated component.

At the time of document export, the archive document ID is passed downstream through the Intelligent Capture URN field.

Late archiving integration

Late archiving means that the image is to be archived after processing in Intelligent Capture.

You can configure PIC for Transcripts to produce a TIFF or PDF of the image and store the image in the Visibility reporting database.

What is solution reporting?

PIC for Transcripts contains connectors to populate the Visibility reporting tables.

Install PIC for Transcripts

What is the PIC for Transcripts installation and setup process?

The following steps outline the high-level procedures that you need to perform to install and configure PIC for Transcripts.

1. Install PIC for Transcripts manually.
2. Install the Transcript Table Extraction engine.
3. Configuration database and INI file settings.
4. Configure data export.
5. Set up and add clients.
6. Create institution or applicant master partitions, or create identification fields without using a partition.
7. Set up business rules relating to predefined data fields and document types.
8. Activate the Dynamic Verifier Form.
9. Set up users.
10. Set up review states.
11. Create a connection to Perceptive Visibility.
12. Install and configure the PIC for Transcripts Connector. For details about the PIC for Transcripts Connector, refer to the Perceptive Transcript File Drop Connector 6.7 Installation Guide available in the Product Documentation tab in the Customer Portal on the Perceptive Software website.

About preparing to install PIC for Transcripts

Before you install PIC for Transcripts, verify that you have completed the following tasks.

- Complete the recommended registry and configuration settings.
- Install Perceptive Intelligent Capture 5.5 with service pack 2
- Obtain and install a Perceptive Intelligent Capture license file. This file is located in the <*InstallDir*>\Components\Cairo directory.
- Obtain the PIC for Transcripts installation files.

Download the installation files

To obtain the PIC for Transcripts installation files, complete the following steps.

1. Go to the Perceptive Software website at www.perceptivesoftware.com and log in to the Customer Portal.
2. In the **Product Downloads** page, search for the one of the following files that matches your version.

   • Perceptive Intelligent Capture for Transcripts 1.x.zip
3. Download the relevant files to a temporary directory on your computer.

About manually installing PIC for Transcripts

Installing PIC for Transcripts manually is comprised of the following steps.

1. Install and configure the transcript table extraction.
2. Configure the registry key settings.
3. Configure the **Verifier Thin Client** dialog box settings.
4. Create the PIC for Transcripts folder structure.
5. Install the PIC for Transcripts project files.
6. Create the Pool directory.
7. Run the database table creation script.

About configuring the registry key settings

The following registry settings are required when installing PIC for Transcripts.

- AnalyzeLinesOptionally
- ASEnginePoolAllowedCharDifference

AnalyzeLinesOptionally provides optimal performance when the system is calculating the orientation of each text line from the OCR results.

ASEnginePoolAllowedCharDifference is recommended for projects where applicant or institution identification using the associative search engine is being used. By default, Intelligent Capture filters out near-identical applicant or institution, which is often undesirable. Adding a registry key ensures that the system returns all relevant information.

Configure the registry key settings

To set the registry keys, complete the following steps.

1. Open the **Windows registry editor**.
2. If it does not exist already, in **HKEY_LOCAL_MACHINE\SOFTWARE\Perceptive**, add a new folder and name it Cedar.
3. In the new Cedar folder, create the AnalyzeLinesOptionally and ASEnginePoolAllowedCharDifference keys as **DWORD** values.
4. Set both keys to a value of `1` with a hexadecimal base.

Configure dialog box settings for the Verifier Thin Client

If you are deploying the Verifier Thin Client, then you need to update the web.config file so that the optional dialog boxes and information and are activated within the Verifier interface.

To configure the web.config file, complete the following steps.

1. Navigate to **<*InstallDir*>\Perceptive\Perceptive Intelligent Capture Web Server**.
2. Open the **web.config** file.
3. Set the following parameters to `True`.

   ```
   <mouseClicked enabled="True"/>
   <tabPressed enabled="True"/>
   ```

4. Save and close the file.

Create the PIC for Transcripts folder structure

The first step in installing PIC for Transcripts is to create the underlying folder structure within Windows Explorer. To create the folder structure, complete the following steps.

1. Create a folder directory on the hard drive, for example `C:\PSWProjects\ICT.`
2. Within this directory, create four folders and name them as follows.
   - Import
   - Batch
   - Global
   - Export

Install PIC for Transcripts project files

To install the PIC for Transcripts project files, complete the following steps.

1. Navigate to the **Global** directory created in step 2 in the previous section.
2. Copy the following files from the installation directory to the **Global** directory.
   - The **SDP** file
   - All contents of the **Train** directory.
   - The **INI** file
3. Rename the **SPD** and **INI** files to something appropriate for your institution. The SPD and INI files should have the same name, but ensure that you retain the SDP and INI file extensions. For example, stateuniversity.sdp and stateuniversity.ini.

   Note In this document, the INI file is referred to as <*project*>.ini.

Create the Pool directory

The pool directory stores the internal representation of the applicant and institution master data, which is accessed by the system at runtime when the document applicant and institution are being determined. To create the pool directory, complete the following step.

- In the **Global** directory, create a new directory and name it `Pool`.

Run the database table creation scripts

It is recommended that the PIC for Transcripts tables exist in their own database. To create the database and PIC for Transcripts tables, complete the following steps.

1. On your Microsoft SQL Server, create a new database.
2. Run the **MasterICT.sql** script against the new database. This script is included in the PIC for Transcripts installation package.
3. Run each update script against the newly created database in the following order.
    1. 1210To1300.sql
    2. 1300To1310.sql
    3. 1310To1312.sql
    4. 1312To1330.sql

What is the Transcript Table Extraction (TTE) engine?

PIC for Transcripts utilizes an expanded version of the Phoenix Table Extraction engine, which has been optimized for college transcript coursework extraction. The Transcript Table Extraction (TTE) engine is not installed with Intelligent Capture by default and must be installed manually.

Install and configure the Transcript Table Extraction (TTE) engine

To install the TTE engine, complete the following steps.

1. Stop all Intelligent Capture applications and services.
2. Copy the contents of **TTE_AddOn_B5556d.zip** into the <InstallDir>\Perceptive\Components\Cedar directory.
3. Navigate to **<InstallDir>\Perceptive\Components\Cedar** and run the **RegCdr.bat** file as an administrator.
4. Execute one of the following registry files for your operating system.

· For a 32-bit operating system, execute the **TTE_Registry_32.reg** file.

· For a 64-bit operating system, execute the **TTE_Registry_64.reg** file.

About configuring the INI file

The final step to completing the project installation is to perform a basic configuration of the <*project*>.ini file. This is the INI file configuration for the installation of PIC for Transcripts. You need to configure other parameters in the INI file after you have finished the installation.

Configure the INI file

To perform a basic configuration of the INI file, complete the following steps.

1. Navigate to the **Global** directory where the PIC for Transcripts project file is installed.
2. Open the **<*project*>.ini** file.
3. Configure the settings in the following table.
4. Save and close the INI file.

Setting	Explanation of usage
GRL_VL_ProjectName	This column represents the name of the project, which is recorded in the Visibility reporting database. It is customary to set this value to the name of the end user. For example, `GRL_VL_ProjectName=Perceptive`.
GRL_VL_Version (Optional)	This is the version number of the project, which is recorded in the Visibility reporting database. This can be set to `1`, but can be left blank or changed to assign document records to different testing cycles.
GRL_VL_ClientName (Optional)	This is the default client name that is recorded in the Visibility reporting database for each document that is processed. As it is superseded by settings in the BRWClient table, it is not mandatory to populate this value.
GRL_OP_ReadSettingsFromDB	This controls whether the project reads configuration settings from the database created in the previous step. Set this to `Yes`.
GRL_VL_SQLConnectionGroup	This is the reference to the SQL connection group in the SQL section of the INI file, which contains the connection string to the project database. Set this to `01`.
GRL_OP_BatchInDatabase	This specifies whether the Intelligent Capture document batches are held in the Percpetive_Database database or if they are held as a batch root in the file system. It is recommended that document batches are always held in the database. Set this setting to `Yes`.
GRL_VL_BatchSQLConnection Group	This is the reference to the SQL connection group in the SQL section of the INI file that contains the connection string to the main Perceptive_Database database. Set this to `02`.
SQL_VL_01_ConnectionString	Set this to the connection string for the PIC for Transcripts database. For example, `SQL_VL_01_ConnectionString=Provider=SQLOLEDB.1;Password=secret;Persist Security Info=True;User ID=sa;Initial Catalog=ICT;Data Source=W2008-JOHNSMITH\sqlexpress`

Setting	Explanation of usage
SQL_VL_02_ConnectionString	Set this to the connection string for the main Perceptive_Database database. For example, `SQL_VL_02_ConnectionString=Provider=SQLOLEDB.1;Password=secret;Persist Security Info=True;User ID=sa;Initial Catalog=Perceptive_Database;Data Source=W2008-JOHNSMITH\sqlexpress`

What are the INI file settings?

The INI file is subdivided into sections that control different aspects of the project file behavior. The sections include the following.

- GRL
- IMP
- REP
- SQL
- ASA

What is the INI file nomenclature?

Each file setting parameter is comprised of one of the following nomenclatures.

- XXX_YY_DDDDD=ZZZ
- XXX_YY_NN_DDDDD=ZZZ

Where

- XXX is the INI file section ID, such as GRL, IMP, REP, SQL, and ASA.
- YY is the type of setting where VL denotes a value or list of values, and OP denotes an on/off switch and is set either to Yes or No.
- NN is an optional INI file group ID used to tie multiple individual settings together to form a settings group. This is similar to a database table where XXX is table name, NN represents the unique table row and DDDDD represents the unique table column name.
- DDDDD is the parameter name, which may be more or less than five characters.
- ZZZ is the parameter setting, which can be completed by the individual configuring the project and can be more or less than three characters. Only ZZZ values should ever be changed in the file, though additional NN settings groups may also be added as appropriate.

What is the GRL section in the INI file?

This section contains global settings for the project that are used for the purposes of solution reporting.

The following parameters can be set.

Parameter	Type	Description
ProjectName	Freetext	This is the name of the project.
Version	Number	This is the version number of the project implementation at the client.
ClientName	Freetext	This is the name of the client.
VerifierFormStyle	Freetext	Color scheme applied to the Verifier form. The options are as follows. SAP – SAP style color scheme If any other setting is applied (including blank), the system displays the default Verifier color scheme, such as gray form with valid fields marked in green and invalid fields marked in red.
UseDynamicVerifierForm	Yes/No	This is the flag to indicate whether the project should use the dynamic verifier form.
ReviewState	Freetext	This is the RTS state that a document is set to if it is subject to review.
ReadSettingsForDB	Yes/No	This is the flag to indicate whether the PIC for Transcripts settings is read from the database. This value should always be set to Yes.
SQLConnectionGroup	NN	This is the numeric reference to the SQL connection group that represents the database in which the configuration tables have been created. This is set to `01`, `02`, and so on.
DynamicDebug	Yes/No	If this is set to `Yes`, logging for the internal mechanics for the layout of the dynamic verifier form is written into the standard PIC for Transcripts Verifier log file.
BatchInDatabase	Yes/No	This is the flag to indicate whether the batch containing the production documents exists within a database or within a batch root folder. This should always be set to `Yes`.

Parameter	Type	Description
BatchSQLConnectionGroup	NN	This is the numeric reference to the SQL connection group that represents the primary PIC for Transcript database. This is set to `01`, `02`, and so on.
ActivateInstitutionFiltering	Yes/No	This is the flag to specify whether the system utilizes multiple sets of institution master data/partitions.
InstitutionFilterColumn	Freetext	This is the case-sensitive name of the database column in the institution master table that contains the partition ID.
ActivateApplicantFiltering	Yes/No	This is the flag to specify whether the system is to utilize multiple sets of applicant master data/partitions.
ApplicantFilterColumn	Freetext	This is the case-sensitive name of the database column in the applicant master table that contains the partition ID.
BufferClientSettings	Yes/No	If set to `Yes`, the system buffers the configuration settings for the current client to reduce repeated calls to the database. For example, if the system is processing a document for client 1, then following an initial read of the database, those settings are stored in memory. If the next document is also for client 1, then the system pulls the settings from memory, rather than read the database. Only settings for the current client are held in memory. This means that any changes made to the database may not take effect instantly for all clients, and a restart of the RTS would be required. If this is not desirable, this option is set to `No`.

What is the IMP section in the INI file?

This section contains settings revolving around document import, specifically the mapping of values contained within the image filename to fields in Intelligent Capture. This provides a simple means to pass data to PIC for Transcripts from an upstream system.

Filename components are separated by a character, such as COMPONENT1.tif, COMPONENT1_COMPONENT2.tif, COMPONENT1_COMPONENT2_COMPONENT3.tif, and so on.

For example, IMP_VL_ScanDate=COMPONENT1 for the 12022008_1234_123456.tif file inserts 12022008 into the ScanDate field in Intelligent Capture.

The following settings are available.

Parameter	Type	Description
URN	Freetext	This is the document unique reference number.
BatchName	Freetext	This is the document batch name.
ScanDate	Freetext	This is the document scan date.
PriorityFlag	Freetext	This is the document priority flag.
TranscriptType	Freetext	This is the document transcript type.
DestinationArchive	Freetext	This is the document destination archive.
InputSource	Freetext	This is the document input source, such as SCAN, EDI, and EMAIL.
ClientID	Freetext	This is the document client ID. In a multi-client project, this value must be mapped. If not, the configuration set associated with client zero is used.
LocationID	Freetext	This is the document Location ID. This field can be used to hold the operation location ID that is relevant for the document. For example, the ID of a shared service center. The value contained within the filename is written into the Visibility reporting tables for that particular document to enable location level reporting.
PriorityFlagYes	Freetext	This is the value that denotes a positive setting for the priority flag.
DateFormat	Freetext	This is the format of a date contained within the document filename. Options are DDMMYYYY, MMDDYYYY or YYYYMNDD.
ApplicantID	Freetext	This is the applicant ID. This field can be used to pass a known applicant ID to the system so that it is used instead of performing a search of the document.

Parameter	Type	Description
SiteID	Freetext	This is the site ID. This field can be used to pass a site Id to the system for students that have multiple addresses.
ComponentSeparator	Freetext	This is the character that is used to separate the different components of a file name.

What is the REP section in the INI file?

This section contains the configuration settings relating to Visibility reporting.

The following parameters can be set.

Parameter	Type	Description
ConnectToReportingDB	Yes/No	This is the flag to set whether the project writes out reporting data or not.
SQLConnectionGroup	NN	This is the SQL connection group specifying the reporting database connection string as set in the SQL section. If no connection group is specified, the system uses group 01.
ReportingInDesigner	Yes/No	This is the flag to indicate whether documents processed or analyzed in the Intelligent Capture Designer Module should have the results written to the reporting database.
StartNewRecordFor ImportedDocument	Yes/No	If this is set to Yes, Intelligent Capture creates a new reporting record for each document imported into Designer, removing any old ones for the same document key. If this is set to No, Intelligent Capture only writes to the reporting database if an entry exists for the same document key. This can be used in the event that the reporting trail begins at the scan station.
ReportingDBDocument Table	Freetext	Name of the document header table in the reporting database
ReportingDBFieldTable	Freetext	This is the name of the document header table in the reporting database.
ReportingDBLineItemsTable	Freetext	This is the name of the document field table in the reporting database.
ReportingDBLineItemsTable2	Freetext	This is the name of the military coursework line items table in the reporting database.
ReportingDBHistory Table	Freetext	This is the name of the document history table in the reporting database.

Parameter	Type	Description
ReportingDBImage Table	Freetext	This is the name of the document image table in the reporting database.
StoreImageInReporting Tables	Yes/No	This indicates whether the document image is stored in a binary type field in the reporting database.
ReportingKey	Freetext	This contains the component to be used as the database table key for the document record. If left blank, the key is set to the image filename (minus the file extension). If just a component of the filename is required, then this value is populated with `URN`, then the URN component of the filename is mapped correctly in the IMP section.
ArchiveURL	Freetext	This contains the mask for the URL associated with a document link. XXXXX should denote the part of the URL that is substituted with the unique document ID from the point of view of the archiving system to form a valid URL that retrieves the document.
StorageDirectory	Freetext	This is the path to the directory that is used as a repository to store images subsequent to document export.

What is the SQL section in the INI file?

This section contains the SQL connection strings that are used by Intelligent Capture.

The solution supports Oracle and Microsoft SQL Server databases.

Parameter	Type	Description
NN_ConnectionString	Freetext	This is the connection string for SQL group NN.

What is the ASA section in the INI file?

This section contains settings that control the Associate Search Engine (ASE) pools used for the institution and applicant look-ups in Intelligent Capture.

The following settings are configurable.

Parameter	Type	Description
Class	Freetext	This is the name of the Intelligent Capture class on which the field was created.
Fieldname	Freetext	This is the technical name of the Intelligent Capture field.
AlphaNum	Yes/No	This indicates whether the key field for the pool record is alphanumeric if set to `Yes`. If set to `No`, the field is assumed to be numeric. This must be set correctly to generate the pool correctly.

Parameter	Type	Description
PoolRelative	Yes/No	This indicates whether the location of the pool directory is relative to the project file.
PoolPath	Freetext	This is the UNC path to the pool directory if it is not relative to the project file.
PoolDirectory	Freetext	This is the name of the pool directory.
PoolName	Freetext	This is the name of the pool.
FileRelative	Yes/No	This indicates whether the location of the pool import CSV file is relative to the project file.
ImportPathFilename	Freetext	This is the UNC path to the pool import CSV file if it is not relative to the project file.
ImportFilename	Freetext	This is the name of the pool CSV import file.
ImportODBCDSN	Freetext	This is the name of the user DSN for the ODBC pool import.
ImportODBCSelect	Freetext	This is the select statement used to create the pool.
ImportODBCUser	Freetext	This is the User ID used to connect to the database. This can be left blank and specified in the PIC for Transcripts project file if security requires it.
ImportODBCPWD	Freetext	This is the user password to access the database. This can be left blank and specified in the Intelligent Capture project file if security requires it.
AutoImportOption	FILE, NoNE or ODBC	This indicates the source from which the pool is created via the Intelligent Capture Runtime server (RTS). If set to None, the pool is not updated automatically by RTS.
FirstPageOnly	Yes/No	This is the flag to indicate whether only the OCR text on the first page of the document is used to determine the field result.
PageZoneALeft	0-100	This is Zone A left search %.
PageZoneAWidth	0-100	This is Zone A width search %.
PageZoneATop	0-100	This is Zone A top search %.
PageZoneAHeight	0-100	This is Zone A height search %.
PageZoneBLeft	0-100	This is Zone B left search %.
PageZoneBWidth	0-100	This is Zone B width search %.
PageZoneBTop	0-100	This is Zone B top search %.
PageZoneBHeight	0-100	This is Zone B height search %.

What are the PIC for Transcript database tables?

The PIC for Transcripts database is subdivided into a series of tables, each containing settings that relate to various aspects of the solution.

Some of the tables contains settings that you can configure to meet your institution's transcript extraction requirements.

What is the BRWAPT table?

This table contains settings for validating an extracted applicant ID.

Parameter	Type	Description
ProfileID	Integer	This is the profile ID.
ValidateFromASSA	Boolean	This denotes whether an extracted applicant ID is validated against the Associative Search Engine Pool/Applicant Extract. It is recommended that this setting should always be set to `True`.
AlphNumSiteSeparator	Freetext	This is the special character used to separate an applicant ID and site ID in the unique ID column in the applicant ASE pool.
DefaultCountry	Freetext	If no country column is available in the applicant extract used by the ApplicantASE field or the value in the country column is blank, a default country for all applicants may be specified here. This is a two-character ISO-code, such as United States = US, United Kingdom = GB, Germany = DE, and so on.
Weight	Integer	If ValidateFromASSA is set to true, this is the minimum weight for an Applicant candidate to be valid.
Distance	Integer	If ValidateFromASSA is set to true, this is the minimum distance for an Applicant candidate to be valid.

What is the BRWAPTPartition table?

This table contains a list of the applicant partitions active within the project. An applicant partition must be registered within this table before it can be assigned to a client.

Parameter	Type	Description
ApplicantPartition	Integer	This is the unique ID of the applicant partition.
Description	Freetext	This is the description of the partition.

What is the BRWClient table?

This table is where clients are set up and configured.

Parameter	Type	Description
ClientID	Integer	This is the unique ID of the client, which must always be set to an integer value.
ProfileID	Integer	This is the ID of the profile assigned to the client. The profile controls what fields are extracted and how they are validated. More than one client may share the same profile ID if the extraction and validation requirements are identical.
ExportProfileID	Integer	This is the ID of the export profile assigned to the client. The export profile ID controls how data is exported for a client. More than one client may share the same export profile ID if the export requirements are identical.
ClientName	Freetext	This is the free text string containing the name of the client. This data is written into the Visibility reporting database for each document assigned to a client.
InstructionsProfileID	Integer	Not used in current release.
ForceVerify	Boolean	This is a flag value that controls whether all documents for this client should stop in Verifier. If set to `True`, all documents will stop. If set to `False`, only documents requiring user attention will stop.
ClientGroup	Integer	This is the ID of the client group to which the client belongs. It is an integer value that can be set freely by the system administrator. The client group is the means by which users are assigned to have access to documents belonging to specific clients.
RequiresReview	Boolean	This is a Boolean flag that indicates whether documents assigned to the client should always be subject to review post verification.
InstitutionPartition	Integer	This is the ID of the institution master data partition to be used by the client.
ApplicantPartition	Integer	This is the ID of the applicant master data partition to be used by the client.
Priority	Integer	When documents are imported into PIC for Transcripts, they are placed into batches and each batch is assigned a priority. This priority controls the order by which the RTS component of PIC for Transcripts processes the batches, and also the order in which the documents appear in the Verifier application. The priority scale runs from 1 to 9, with 1 having the highest level of priority. If this field is populated with 1, it means that all batches containing documents from this client is accorded a priority of 1.

What is the BRWCRS table?

This table contains the validation options available for the course number field for College transcripts.

The following settings are available.

Parameter	Type	Description
ProfileID	Integer	This is the profile ID.
ValidateFromDB	Boolean	This is the flag to denote whether the course number is validated against course numbers from the same institution in a database table.
SQLConnectionGroup	Integer	This is the SQL connection group specifying the course number validation database connection string as set in the SQL section. If no connection group is specified, the system uses group 01.
CourseTable	Freetext	This is the name of the course number validation database table.
InstIDColumn	Freetext	This is a column in the database table that holds the institution ID. This setting is mandatory.
SiteID	Freetext	This is a column in the database table that holds the institution's site ID if applicable.
CourseIDColumn	Freetext	This is a column in the database table that holds the course ID. This column must be a combination of the course subject and course number, for example MTH101.
DescColumn	Freetext	This column in the database table that holds the course description.
UseInternalIstID	Boolean	This is a flag to denote whether the internal SIS institution ID is used when validating the course number.
IncludeSite	Boolean	This is a flag to denote whether the institution's Site ID is used when validating the course number.
RemoveInstLeadZeros	Boolean	This is a flag to denote whether leading zeroes is removed from the institution ID when validating the course number.
UpdateDBAtExport	Boolean	If set to `True`, the system updates the course number validation table with new course numbers from the current document at the point of document export.
SubjectColumn	Freetext	This is a column in the database table that holds the course subject.
CourseNumberColumn	Freetext	This is a column in the database table that holds the course number.

What is the BRWDAT table?

The settings in this table control the formatting and validation of the transcript dates.

The following settings are available.

Parameter	Type	Description
ProfileID	Integer	This is the profile ID
VerifierOutputFormat	DDMMYYYY or MMDDYYYY	If set to `DDMMYYYY`, PIC for Transcripts displays the date in Verifier as DD/MM/YYYY. If set to `MMDDYYYY`, PIC for Remittance displays the date in Verifier as MM/DD/YYYY. If set to `YYYYMMDD`, PIC for Remittance displays the date as YYYY-MM-DD.
ExportFormat	MMDDYYYY DDMMYYYY YYYYMMDD	This is the output date format for export. This setting applies to database output and all flat file exports.
ExportSeparator	Freetext	This is the separator that is used when exporting a date value. For example, a slash (/), dot (.), or hyphen (-).
MMDDCountries	Freetext	This is the comma-separated list of countries that use MM/DD/YYYY as the date format preference.
YYMMDDCountries	Freetext	This is the comma-separated list of countries in which YY-MM-DD is a standard date format, for example Sweden. If the country of origin is included in this list and the date is read as 12-01-11, this is formatted as 11/01/2012 (DDMM) or 01/11/2012 (MMDD) if the date year is either the current year, the previous year, or the following year. If the date year is something else, the format is assumed to be DD-MM-YY. Use caution when adding countries to this list.

What is the BRWERR table?

This table contains the list of error message that may be displayed to a user in Verifier, or written into the Intelligent Capture log file. This is a global table that is independent of the client or profile. You can add new error messages using error number 900 onwards.

The following settings are available.

Parameter	Type	Description
ErrorNumber	Integer	This is an error message number.
Message	Freetext	This is the error message text.

What is the BRWEXP table?

This table holds configuration settings relating to the ICT data export options. It is keyed upon the export profile ID, which can be assigned to individual clients.

The following settings are available.

Parameter	Type	Description
EXPProfileID	Integer	This is the export profile ID.
Description	Freetext	This is the description of export profile
RedoAllExports	Boolean	If set to True, the system carries out all export options that have been activated even if that export has been carried out before. For example, if three export options are activated, two are completed and the last one fails, then the document goes to status 750 denoting an export failure. If the flag is set to `False`, upon retrying, only the failed export is carried out. If the flag is set to `True`, then all three exports is performed again.
DefaultExportPath	Freetext	This is the UNC path to the export directory which is used as the default should no export directory be set in RTS.
OutputTiffFile	Boolean	If set to `True`, the system outputs a TIFF file of the document image in the export directory.
Tiffname	Freetext	This setting controls the name of the output Tiff file. If set to `URN`, this names the file according to the component of the image filename mapped in the IMP section. If left blank, or set to anything else, the filename is set to the same name as the document filename.
TiffDPI	Number	This specifies the DPI of the outputted TIFF image, such as 300. The default TIFF resolution is 300 dpi.
TiffFormat	Freetext	This is the compression format of the outputted TIFF file. The following options are available. **G4FAX** Grade 4 compression **G3FAX** Grade 3 compression **LZWFAX** LZW Compression **HUFFAX** HUF Compression The default compression is G4FAX.
OutputPDF	Boolean	If set to `True`, the system outputs a searchable PDF file for each document.

Parameter	Type	Description
PDFName	Freetext	This setting controls the name of the output PDF file. This can be set to `URN`, which names the file according to the component of the image filename mapped in the IMP section. If left blank or set to anything else, the filename is set to the same name as the document filename.
CustomExport	Boolean	This is a flag to indicate whether a custom export is carried out, as specified in the UserExitCustomExport script user exit.
OutputXMLFile	Boolean	If set to `True`, the system outputs an XML file to the export directory configured on the RTS export instance. If no directory is configured, then the default export path parameter is used. If this is not configured either, then the XML export fails and the batch is sent to status 750.
XMLFilename	URN [blank]	This setting controls the name of the XML output file. This can be set to `URN`, which names the file according to the component of the image filename mapped in the IMP section. If left blank or set to anything else, the filename is set to the same name as the document filename.
XMLFileType	Freetext	This is the file extension applied to the XML file, for example XML = .XML, TXT = .txt. If left blank, the file extension defaults to XML.
XMLEncodingHeader	Freetext	This is the XML file coding header that forms the first line in the XML file. For example, setting the value as `<xml version="1.0" encoding="UTF-16"?>` produces an XML file that supports non-Western characters such as letters from the Russian, Greek and Chinese alphabets.
XMLFileHeader	Freetext	This denotes the value of the file header tag in the XML file, for example <MyFileHeader>. This value defaults to PIC for Transcripts Document if nothing else is set.
XMLFileHeaderAttributes	Freetext	This contains any attributes that need to be assigned to the XML file header tag.
XMLDocName	Freetext	This denotes the value of the tag marking the document name section in the XML file, for example <FileInfo>.
XMLType	Freetext	This denotes the value of the tag marking the transcript type (College, High School, or Military) item in the XML file, for example <type>.

Parameter	Type	Description
XMLStudentHeader	Freetext	This denotes the value of the tag marking the student information section in the XML file, for example <StudentRecord>. This value defaults to StudentData if nothing else is set.
XMLUniversityHeader	Freetext	This denotes the value of the tag marking the university degree summary section in the XML file, for example <universitySummary>. This value defaults to UniversitySummary if nothing else is set.
XMLUniversityDegreeHeader	Freetext	This denotes the value of the tag marking the degree section in the XML file, for example <universityDegree>. This value defaults to Degree if nothing else is set.
XMLUnivRecordHeader	Freetext	This denotes the value of the tag marking the university record section in the XML file, for example <universityInstitutionalRecord>. This value defaults to UniversityRecord if nothing else is set.
XMLHighSchoolRecordHeader	Freetext	This denotes the value of the tag marking the high school record section in the XML file, for example <highschoolInstitutionalRecord> This value defaults to HighSchoolRecord if nothing else is set.
XMLHighSchoolHeader	Boolean	This denotes the value of the tag marking the high school header field section in the XML file, for example <HSSummary>. This value defaults to HighSchoolSummary if nothing else is set.
XMLIssueDate	Freetext	This denotes the value of the tag marking the transcript issue date in the XML file, for example <DateIssued>.
XMLUnivCourseHeader	Freetext	This denotes the value of the tag marking the university coursework section in the XML file, for example <universityCourseRecords> This value defaults to UniversityCourses if nothing else is set.
XMLUnivCourseTag	Freetext	This denotes the value of the tag marking each individual college course in the XML file, for example <Course>.
XMLMilCourseTag	Freetext	This denotes the value of the tag marking the military coursework section in the XML file, for example <militaryCourseRecords>. This value defaults to MilitaryCourses if nothing else is set.
EXPFiltering	Boolean	This is a flag to indicate that whether the exported courses is filtered. If set to `True`, the exported courses is filtered based on the value of EXPExcludedGradeLetters column. Otherwise, the exported courses is not filtered.

Parameter	Type	Description
EXPExcludedGradeLetters	Freetext	Comma separated list of grade letters that is filtered out from export files. This setting only applies if the EXPFiltering field is set to `True`. If the value of the field is set to empty string, no course is filtered out.

What is the BRWEXPMilCourses table?

This table is used to map the military coursework fields into fields in the XML file or columns in a database.

The following settings are available.

Parameter	Type	Description
EXPProfileID	Integer	This is the export profile ID.
FieldName	Freetext	This is the name of the field.
XMLTag	Freetext	This column represents that tag that is used for the field in an exported XML file. If left blank, the field is not exported.
DBColumnName	Freetext	This is the technical name of the target field in the export database.

What is the BRWEXPStudentHeader table?

This table is used to map the applicant header export fields into fields in the XML file or columns in a database.

The following settings are available.

Parameter	Type	Description
EXPProfileID	Integer	This is the export profile ID.
FieldName	Freetext	This is the name of the field.
XMLTag	Freetext	This column represents that tag that is used for the field in an exported XML file. If left blank, the field is not exported.
DBColumnName	Freetext	This is the technical name of the target field in the export database.

What is the BRWEXPUnivCourses table?

This table is used to map the university coursework fields into fields in the XML file or columns in a database.

The following settings are available.

Parameter	Type	Description
EXPProfileID	Integer	This is the export profile ID.
FieldName	Freetext	This is the name of the field.
XMLTag	Freetext	This column represents that tag that is used for the field in an exported XML file. If left blank, the field is not exported.
DBColumnName	Freetext	This is the technical name of the target field in the export database.

What is the BRWEXPUnivHeader table?

This table is used to map the university header fields into fields in the XML file or columns in a database.

The following settings are available.

Parameter	Type	Description
EXPProfileID	Integer	This is the export profile ID.
FieldName	Freetext	This is the name of the field.
XMLTag	Freetext	This column represents that tag that is used for the field in an exported XML file. If left blank, the field is not exported.
DBColumnName	Freetext	This is the technical name of the target field in the export database.

What is the BRWEXPUnivDegree table?

This table is used to map the university degree fields into fields in the XML file or columns in a database.

The following settings are available.

Parameter	Type	Description
EXPProfileID	Integer	This is the export profile ID.
FieldName	Freetext	This is the name of the field.
XMLTag	Freetext	This column represents that tag that is used for the field in an exported XML file. If left blank, the field is not exported.
DBColumnName	Freetext	This is the technical name of the target field in the export database.

What is the BRWEXPUnivRecord table?

This table is used to map the institution export fields into fields in the XML file or columns in a database.

The following settings are available.

Parameter	Type	Description
EXPProfileID	Integer	This is the export profile ID.
FieldName	Freetext	This is the name of the field.
XMLTag	Freetext	This column represents that tag that is used for the field in an exported XML file. If left blank, the field is not exported.
DBColumnName	Freetext	This is the technical name of the target field in the export database.

What is the BRWFLD table?

This table controls which fields are activated for a given profile, along with their corresponding types and validation settings.

The following settings are available.

Parameter	Type	Description
ProfileID	Integer	This is the profile ID.
FieldName	Freetext	This is the name of the PIC for Transcripts field. Standard field names should not be altered.
VerifierLabel	Freetext	This is the text indicating how the field is labeled on the dynamic Verifier form.

Parameter	Type	Description
Active	Boolean	This is the flag to indicate whether the field is activated for the profile.
RequiredInRTS	Boolean	This is the flag to indicate whether the field is mandatory in RTS.
RequiredInVerifier	Boolean	This is the flag to indicate whether the field is mandatory in Verifier.
CountryFilter	Freetext	This is the comma-separated list of country ISO codes that allows fields to be mandatory only for specific countries. This setting is used in conjunction with RequiredInRTS and RequiredInVerifier. If a field is set for mandatory for either of them and the vendor country of origin is not specified in the list, then the field reverts to being optional.
FieldType	AMOUNT DATE TEXT TABLE	This is the type of the field.
ForceVerify	Boolean	This is the flag to indicate whether the field should always be marked as invalid and sent to a Verifier for review.
DefaultValue	Freetext	This is the field default value.
DefaultIfNothingExtr	Freetext	This is the default value if the system does not automatically capture a value from the document.
SubRule	Integer	This is the field substitution rule for text fields as registered in the BRWSUBSTITUTION table.
MinLength	Integer	This is the field minimum permitted length for text fields.
MaxLength	Integer	This is the field maximum permitted length for text fields.
RightJustify	Boolean	If a value has been entered in PadChar, and this parameter is set to `True`, the system pads the field value with that character from the left until the length specified in MaxLength is reached. If a value has been entered in PadChar, and this parameter is set to `False`, the system pads the field value with that character to the right until the length specified in MaxLength is reached.
PadChar	Single character	This is the padding character for a text field.
RemoveAllSpecials	Boolean	This is the flag to indicate whether all special characters is removed from a text field.

Parameter	Type	Description
RemoveBlanks	Boolean	This is the flag to indicate whether spaces is removed from a text field.
KeepCertainSpecials	Freetext	This is the non-comma-separated list of special characters that is retained if RemoveAllSpecials is set to True.
RemoveStartEnd	Boolean	If set to `True`, PIC for Transcripts removes any special characters at the beginning and at the end of an extracted text value.
SubstringStartPos	Integer	The starting character used in conjunction with SubstringLength when trimming an extracted value. Positive numbers start from the left while negative numbers start from the right.
SubstringLength	Integer	This is the substring length.
RemoveLeadingZeros	Boolean	If set to `True`, PIC for Transcripts removes any leading zeroes from an extracted text value.
DecimalPlaces	Integer	This is the number of decimal places for an amount field that is outputted at time of export.
NegativeType	Integer	This integer setting controls the output during export if the extracted value for an amount field is less than zero. Possible settings and their effects are as follows. **1** The minus sign to appear after the amount, such as 100.00- **2** The minus sign to appear before the amount, such as 1 -100.00 **3** Is the value to appear in parentheses, such as 1 (100.00)
OutputForZero	Freetext	This is the output value during export for an amount value if it is zero.
SubstituteValueIfOverZero	Freetext	This is the output value during export for an amount value if it is greater than zero.
FutureDays	Integer	This numerical value indicates the number of days in the future from the present date that an extracted date may be considered to be valid. For example, if today's date is March 20th and a date is extracted as March 31st, and the value is set to `10`, then the system marks the field invalid as the extracted date is 11 days in the future. If future dates are not permitted, then the column value is set to `0`. To disable the check entirely, set the column value to `-1`. In Verifier, the user may pass any value as long as it is a valid date.

Parameter	Type	Description
NoDaysInPast	Integer	This numerical value indicates the number of days in the past back from the present date that an extracted date may be considered to be valid. For example, if today's date is March 20th and a date is extracted as March 9st, and the value is set to `10`, then the system marks the field invalid as the extracted date is 11 days in the past. If past dates are not permitted, then the column value is set to `0`. To disable the check entirely, set the column to `-1`. In Verifier, the user may pass any value as long as it is a valid date.
DateOnlyInCurrentMonth	Boolean	If this column is set to True, then an extracted date stops in Verifier if the date is not in the current month. In Verifier, the user may pass any value as long as it is a valid date.
FieldMask	Freetext	This is the comma-separated list of valid entries for the extracted or user-entered value. For example, if the content of this column is set to `ABCD,WXYZ` then no value is permitted unless it is equal to either ABCD or WXYZ. Wildcard characters are also permitted, where a hash symbol (#) is used to represent any number, an at symbol (@) is used to represent any letter, and a question mark (?) is used to represent either a number of a latter. For example, if entry is restricted to being 10 followed by a letter then a hyphen and then five digits, the value `10?-#####` is entered into the field.

What is the BRWGRD table?

This table contains settings for grade validation for college transcripts.

The following settings are available.

Parameter	Type	Description
ProfileID	Integer	This is the instructions profile ID.
AcceptedGradeLetters	Freetext	This is a comma-separated list of valid entries for the extracted grade.
GPAValidationTolerance	Decimal	This is the tolerance that is acceptable between the Calculated GPA derived from the extracted coursework and the Cumulative GPA. This setting only applies to College transcripts.
ExcludedGradeLetters	Freetext	This is a comma-separated list of grades to exclude when calculating the GPA. This setting only applies to College transcripts.

Parameter	Type	Description
EnableGPACalcValidation	Boolean	When set to `True`, the extracted GPA is compared against the GPA calculated from each course on the transcript.
AllowInvalidGPAInVerifier	Boolean	When set to `True`, this allows a verifier to manually validate the Cumulative GPA field.

SetInvalidReasonInVerifier	Boolean	When set to `True`, this translates the following invalid reasons.
		`NoNE ' CUMULATIVE GPA <> CALCULATED GPA`
		`APPLICANT NoT FOUND ' GPA AND APPLICANT INVALID`
		`INSTITUTION NoT FOUND ' GPA AND INSTITUTION INVALID`
		`INSTITUTION AND APPLICANT NoT FOUND ' GPA, INSTITUTION, APPLICANT INVALID`

AcceptedHSGradeLetters	Freetext	This is a comma-separated list of valid entries for the extracted grades of a high school document.
AcceptedHSNumericGrade Range	Freetext	This is a range of valid numbers for the extracted grades of a high school document.

What is the BRWINF table?

This table controls setting associated with the Verifier information dialog boxes. It is a global table independent of the client or profile. This table is not used in the current release.

Parameter	Type	Description
DialogHeader	Freetext	Information box title bar text

What is the BRWINFType table?

This table stores the information messages that appear in Verifier. It is a global table independent of the client or profile. This table is not used in the current release.

Parameter	Type	Description
InfNumber	Integer	Information message ID
Message	Freetext	Information message for the message ID.

In common with table BRWERR, only the text associated with each message is changed.

What is the BRWINS table?

This table contains settings for validating an extracted institution ID.

The following settings are available.

Parameter	Type	Description
ProfileID	Integer	This is the profile ID.
ValidateFromASSA	Boolean	This denotes whether an extracted institution ID is validated against the Associative Search Engine Pool / Institution Extract. Set this to `True`.
AlphNumSiteSeparator	Freetext	This is the special character used to separate an institution ID and site ID in the unique ID column in the institution ASSA pool.
DefaultCountry	Freetext	If no country column is available in the applicant extract used by the ApplicantASE field or the value in the country column is blank, a default country for all applicants may be specified here. This is a two-character ISO-code, for example United States = US, United Kingdom = GB, Germany = DE.
DefaultMilitaryInsID	Freetext	This is the default institution ID to be used on all military transcripts. If this column is populated, the institution information for all branches of military transcripts uses the populated value. Installations that utilize a site ID for the institutions should include the site ID and site ID separator in the value, such as 12345-1. The ID entered must exist in the Institution ASE pool. If this column is blank/NULL, PIC for Transcripts utilizes the standard associative search to match the institution ID within the Institution ASE pool.

What is the BRWINSPartition table?

This table contains a list of the institution partitions active within the project. An institution partition must be registered within this table before it can be assigned to a client.

The following settings are available.

Parameter	Type	Description
InstitutionPartition	Integer	This is the unique ID of the institution partition.
Description	Freetext	This is the description of the partition.

What is the BRWINSTR table?

This table contains a list of instructions and their corresponding texts that are available to be assigned to clients. When using the dynamic verifier form, a button is available to deliver an instructional text to the user to help them with processing documents for a specific client. This table is not used in the current release.

The following settings are available.

Parameter	Type	Description
ProfileID	Integer	These are the instructions Profile ID.
Instructions	Freetext	This is the instructions text.

What is the BRWIVR table?

This section contains the default setting associated with the invalid reason field. This is a global table that works independently of the client or profile.

The following settings are available.

Parameter	Type	Description
DefaultText	Freetext	This is the default invalid reason, such as None.
DefaultExportCode	Freetext	This is the export code associated with the default invalid reason, such as 0.

What is the BRWIVRType table?

This table holds the invalid reasons that may be selected in Verifier. It is a global table that is independent of the client or profile.

The following settings are available.

Parameter	Type	Description
Index	Integer	This is the invalid reason index.
Rule	Freetext	This is the rule ID for the invalid reason. The rule governs how Verifier behaves as a result of a particular invalid reason being selected. The following rules are available. **SETAPPTOVALID** This is the Applicant ID field and Applicant Info fields are set to blank and valid. Verifiers have the ability to enter applicant fields (First Name, Middle Name, Last Name, SSN, and DOB) manually. No applicant ID is exported. **SETINSTTOVALID** This is the Institution ID field and Institution Info fields are set to blank and valid. No Institution ID is exported. **SETINSTANDAPPTOVALID** This is the Institution ID field, Institution Info, Applicant ID, and Applicant Info fields are set to blank and valid. Verifiers have the ability to enter applicant fields (First Name, Middle Name, Last Name, SSN, and DOB) manually. No Institution ID or Applicant ID is exported. New rules can be submitted to Perceptive as a product enhancement request.
VerifierDisplay	Freetext	This is the invalid reason message displayed in Verifier.
ExportCode	Freetext	This is the invalid reason code exported by PIC for Transcripts if the Invalid Reason field is set.

What is the BRWPCO table?

This section contains the connection settings for writing Perceptive Content external messages. This is a global table that works independently of the client or profile.

The following settings are available.

Parameter	Type	Description
WriteExternalMessages	Boolean	This is the flag to indicate whether the document's status is updated in Perceptive Content's external message table.
SQLConnectionGroup	Integer	This is the SQL connection group specifying the external message database connection string as set in the SQL section. If no connection group is specified, the system uses group 01.

What is the BRWProfile table?

This table contains a list of profiles within the project, along with a description. A profile must be registered within this table before it can be assigned to a client.

The following settings are available.

Parameter	Type	Description
ProfileID	Integer	This is the unique ID of the client that must always be set to an integer value.
ProfileDescription	Freetext	This is the description of the profile.

What is the BRWSRA table?

This table holds the mapping between columns in the applicant master data table and the values used internally within the project. The table is global for the project and works independently of the client and profile. The table should only ever consist of a single row.

The following settings are available.

Parameter	Type	Description

ID	Freetext	This is the capture ASE column name denoting the applicant ID.
		For SIS systems where an applicant at a unique address is represented by a combination of the applicant ID and the site ID, the formula for the ID column must be set to the following.
		`Applicant ID * 1000000 + Site ID`
		If the applicant ID or site ID is alphanumeric, the set the formula to the following.
		`ApplicantID~SiteID`
		The delimiter (~ in the above example) is configurable using the AlphNumSiteSeparator parameter in the BRWAPT table. The system raises a configuration error if no delimiter is specified, if it is more than one character, or it does not occur, occurs more than once, or occurs as the first character in the unique ASE ID column.
		The site ID must be mapped to SITEID in the BRWSRA table, and the applicant ID stem must be mapped to EXTERNALID. If the SIS uses an external applicant ID, map this value to EXTERNALID and the internal applicant ID stem can remain unmapped. However, the applicant ID stem component of the ID field is the internal SIS applicant ID.

SiteID	Freetext	This is the Capture ASE column name denoting the applicant site ID. This should only be mapped if the site ID forms part of the ID column above.
FName	Freetext	This is the Capture column name denoting the applicant's first name.
MName	Freetext	This is the Capture ASE column name denoting the applicant's middle name.

Parameter	Type	Description
LName	Freetext	This is the Capture ASE column name denoting the applicant's last name.
Address1	Freetext	This is the Capture ASE column name denoting the first line of the applicant's address.
Address2	Freetext	This is the Capture ASE column name denoting the second line of the applicant's address.
City	Freetext	This is the Capture ASE column name denoting the applicant's city of origin.
State	Freetext	This is the Capture ASE column name denoting the applicant's state.
Zip	Freetext	This is the Capture ASE column name denoting the applicant's zip/postal code.
Country	Freetext	This is the Capture ASE column name denoting the applicant's country of origin.
TelNo	Freetext	This is the Capture ASE column name denoting the applicant's telephone number.
DOB	Freetext	This is the Capture ASE column name denoting the applicant's date of birth.
SSN	Freetext	This is the Capture ASE column name denoting the applicant's social security number.
AppIdentifier	Freetext	This is the Capture ASE column name that represents a unique applicant identifier. For example, SSN.
PartitionID	Freetext	This is the Capture ASE column name that represents the applicant's partition ID.
ExternalID	Freetext	This is the Capture ASE column name denoting the applicant's external ID . If no external applicant ID is used by the SIS but the combination of an applicant ID and a site ID is used to identify a unique applicant address, this column must be mapped to the applicant ID stem.
Custom1	Freetext	This is the Capture ASE column name denoting the custom value that can contain additional information about the applicant.
Custom2	Freetext	This is the Capture ASE column name denoting the custom value that can contain additional information about the applicant.
Custom3	Freetext	This is the Capture ASE column name denoting the custom value that can contain additional information about the applicant.

Parameter	Type	Description
Custom4	Freetext	This is the Capture ASE column name denoting the custom value that can contain additional information about the applicant.
Custom5	Freetext	This is the Capture ASE column name denoting the custom value that can contain additional information about the applicant.

What is the BRWSRC table?

This table holds the mapping between columns in the institution master data table and the values used internally within the project. The table is global for the project and works independently of the client and profile. The table should only ever consist of a single row.

The following settings are available.

Parameter	Type	Description

ID	Freetext	This is the Capture ASE column name denoting the institution ID.
		For SIS systems where an institution at a unique address is represented by a combination of the institution ID and the site ID, the formula for the ID column must be set to the following.
		`Institution ID * 1000000 + Site ID`
		If the institution ID or site ID is alphanumeric, the formula is the following.
		`InstitutionID~SiteID`
		The delimiter (~ in the above example) is configurable via the AlphNumSiteSeparator parameter in the BRWINS table. The system raises a configuration error if no delimiter is specified, if it is more than one character, or it does not occur, occurs more than once, or occurs as the first character in the unique ASE ID column.
		The site ID must be mapped to SITEID in the BRWSRC table and the institution ID stem must be mapped to EXTERNALID. If the SIS uses an external institution ID, map this value to EXTERNALID, and the internal institution ID stem can remain unmapped. However, the institution ID stem component of the ID field is the internal SIS institution ID.

SiteID	Freetext	This is the Capture ASE column name denoting the institution site ID. This should only be mapped if the site ID forms part of the ID column above.
Name	Freetext	This is the Capture ASE column name denoting the institution's name.
Address1	Freetext	This is the Capture ASE column name denoting the first line of the institution's address.
Address2	Freetext	This is the Capture ASE column name denoting the second line of the institution's address.
City	Freetext	This is the Capture ASE column name denoting the institution's city of origin.

Parameter	Type	Description
State	Freetext	This is the Capture ASE column name denoting the institution's state.
Zip	Freetext	This is the Capture ASE column name denoting the institution's zip/postal code.
Country	Freetext	This is the Capture ASE column name denoting the institution's country of origin.
TelNo	Freetext	This is the Capture ASE column name denoting the institution's telephone number.
CEEB	Freetext	This is the Capture ASE column name denoting the institution's CEEB code.
ACT	Freetext	This is the Capture ASE column name denoting the institution's ACT code number.
FICE	Freetext	This is the Capture ASE column name denoting the institution's FICE code.
OPEID	Freetext	This is the Capture ASE column name denoting the institution's OPE ID.
PartitionID	Freetext	This is the Capture ASE column name that represents the institution's partition ID.
ExternalID	Freetext	This is the Capture ASE column name denoting the institution's external ID. If no external institution ID is used by the SIS but the combination of an institution ID and a site ID is used to identify a unique institution address, this column must be mapped to the institution ID stem.

What is the BRWSubstitution table?

This table contains a list of rules for substituting values in an extracted or user-entered text field. Substitution rules are assigned to text fields using the SubRule column in the BRWFLD table.

The following settings are available.

Parameter	Type	Description
SubstitutionRule	Integer	This is the substitution rule ID.
Original	Freetext	This is the segment of a text string to replace.
Replace	Freetext	This is a text string that is used to replace the original value entered in Original if present in the string.

What is the BRWUser table?

Table BRWUser contains a list of active users within the system along with their corresponding authorizations. The table is keyed on a combination of a unique user name and client group.

The following settings are available.

Parameter	Type	Description
UserID	Freetext	This is the Intelligent Capture user ID.
ClientGroup	Integer	This is the client group to which the user has been assigned.
AuthorityLevel	@@@	This is the standard Intelligent Capture role assigned to the user. **VER** This is the standard Verifier user. **SET** This is the Verifier user with permission to change verifier settings. **SLV** This is SET plus the ability to use the supervised learning function **SLM** This is SLV plus the ability to review and promote institution learnsets to the global project. **ADM** This is the administrator.
RequiresReview	Boolean	If set to True, all documents verified by the user go to a review state for quality control.
Domain	Freetext	This is the user Windows domain for Windows based authentication.
Password	Freetext	This is the user password if Windows authentication is not being used.
PrimaryGroupName	Freetext	This is the name of the Web Verifier group to which the user belongs. This column must be populated for both the thick Verifier and Web Verifier installations.

What is the PICT_VersionHistory table?

Table PICT_VersionHistory is a reference table that contains versioning information for the ICT database along with the date of install.

The following settings are available.

Parameter	Type	Description
ID	Integer	This is the ID of version entry.
Version	Integer	This is the database version, such as 1100 – which is build 1100.
InstallDate	Date	This is the date when the database was installed.

What are the PIC for Transcripts customization options?

The PIC for Transcripts solution can be customized to meet specific requirements. Customization takes the following forms.

- Project setting customizations
- Script customizations

Project setting customizations include such things as changing tolerances and thresholds within the project, adding new fields and classes, configuring existing ASE fields as described in Appendix A, changing or creating Verifier forms, or setting up new users.

Existing fields or classes should not be deleted or renamed. Doing so causes the solution to stop operating.

It is also possible to add script customizations into the project. For custom fields and classes, scripts can be added to the appropriate custom field event and class windows. For customizations to the existing Transcripts class, this must be done on the UserExits class script window.

About customizing the project script class

The project script class contains script associated with standard Intelligent Capture system events, which are known as the ScriptModule events.

These ScriptModule events are called at specific points within the Intelligent Capture workflow. For example, at pre and post-import, pre and post-OCR, pre and post classification, and at time of document export.

Do not make any changes to the code on this class level. Doing so causes the solution to stop operating.

About customizing the global variables script class

The global variables script class contains all global variables that are used within the solution. These data definitions are exposed so that you can use them within custom code, and also so that you can see the definition of the custom structures and arrays as a point of reference.

In addition to global variables, this script class also houses a series of common functions and subroutines used throughout the solution. You can use these common functions within custom code placed on the user exit script class, or within code for any additional classes that are created.

The common functions, along with a description of their potential uses, are described in the following table.

Name of function / sub-routine	Description
ReadSettings	This function reads the system configuration INI file.
DicVal	This function returns the value of any parameter contained within the system configuration INI file. The function parameters are as follows. **strKey** This is the name of the INI file parameter. **strDic** This is the name of the INI section in which the parameter is held. Neither strKey nor strDic are case-sensitive. For example, if the EXP_VL_OutputDateFormat configuration parameter contains `MMDDYYYY`, then the following command copies MMDDYYYY into the strOutputDateFormat local string variable. `strOutputDateFormat = DicVal("OutputDateFormat", "EXP")` For configuration parameters that have a Boolean type of OP, the function only returns a value of Yes or No.
Parse_INIVal_Yes	This function receives an INI file strVal parameter value that has been entered against a parameter with Boolean type OP, and determines whether the value is interpreted as Yes or No.

Name of function / sub-routine	Description

SplitString	This subroutine performs a split on a given string based on a nominated separator, and returns the components of the string back to the calling function in an array, along with the number of values in the array.
	The interface parameters are as follows.
	strSource This is the input string to be split.
	strSplitArray This is the array containing the split results passed back to the calling module.
	strDesignator This is the delimiter to be used when performing the split.
	ArrayLineCount This is the number of array elements in the returned strSplitArray.
	For example usage,
	`Dim myString As String`
	`Dim Words() As String`
	`Dim intWordCount As Integer`
	`myString = "MARY HAD A LITTLE LAMB"`
	A space is set as the delimiter.
	`Call SplitString(myString, Words(), " ", intWordCount)`
	The returned Words array would contain the following.
	Words(1) = "MARY"
	Words(2) = "HAD"
	Words(3) = "A"
	Words(4) = "LITTLE"
	Words(5) = "LAMB"
	The returned intWordCount parameter would be set to 5.

fnConvertToExternal	This function converts a date in the date format used internally, for example DD/MM/YYYY, into a specified format.
	The interface parameters are as follows.
	strDate This is the date to be formatted.
	strFormat This is the format of date, either MMDDYYYY or YYYYMMDD. Any other entry returns DDMMYYYY.
	strSeparator This is the separator to be used when converting the date. For example,
	`Dim myDate as string`
	`myDate = "12/08/2009"  12th August 2009`
	`myDate = fnConvertToExternal(myDate, "MMDDYYYY", "-")`
	The value of myDate is set as 08-12-2009.

Name of function / sub-routine	Description

fnConvertToInternal	This function is used to convert a date with a specified format into the date format used internally within PIC for Transcripts, such as DD/MM/YYYY.
	The interface is as follows.
	strDate The date to be formatted to DD/MM/YYYY.
	strFormat The current format of strDate (either YYYYMMDD, MMDDYYYY – any other entry returns DDMMYYYY.
	strSeparator The separator currently used in strDate.
	For example,
	`Dim myDate as String`
	`myDate = "2009-08-12"  12th August 2009`
	`myDate = fnConvertToInternal(myDate, "YYYYMMDD", "-")`
	The value of myDate is now set to 12/08/2009.

fnFormatDateForExport	This function converts a date in the Verifier output format, as configured in the DAT section of the system configuration, into a date in the export output format, as configured in the EXP section. For example, date can appear in any format, but the system converts it to the format specified in the DAT section. If that format is MMDDYYYY, 12th August 2009 is displayed in Verifier as 08/12/2009, which is also the technical content of the field object text property. For example, the contents of pField.Text or pWorkdoc.Fields("MyDate").Text). fnFormatDateForExport takes the technical contents of the field and converts it into the date format as specified in the EXP section. If the export format is YYYYMMDD with a hyphen (-) as the separator, then the following command populates the strDate string variable with 2009-08-12. **strDate** This is the fnFormatDateForExport(pWorkdoc.Fields("MyDate").Text) The interface of the function is as follows. **strDate** This is the date to be converted.

Name of function / sub-routine	Description

fnWriteXMLField	This function writes a single line into the XML file, and is intended for use within UserExitXMLOutput and provides a mechanism to add a custom field into the XML file with a single command.
	
	The interface of the function is as follows.
	
	`strAttribute =` name of the configuration parameter containing the tag for the XML field.
	`strValue =` value of the field to be outputted.
	
	For example, in the configuration database, a new parameter called EXP_VL_XMLHCTranscriptCode has been created with a value of TCODE, and a new field has been created against the Transcript class in the TranscriptCode project, which contains the extracted value of 12345, and this value is written to the document header section of the XML file.
	
	This can be achieved by placing the following in the UserExitXMLOutput framework.
	
	`Select Case strSection`
	`  Case cDefaultXMLDocHeader`
	`  fnWriteXMLField(“TranscriptCode”, pWorkdoc.Fields(“TranscriptCode”).Text)`
	`  Case cDefaultXMLStudentHeader`
	`  …`
	`End Select`
	
	This writes the following line into the document header section of the XML file.
	
	`<TCODE>12345</TCODE>`

fnWriteXMLDateField	This function is used to write out a date field to the XML file where the date to be written is in the Verifier output date format specified in the DAT section of the system configuration. As well as writing the value into the XML file, the system converts the date passed into the date export format as specified in the EXP section of the system configuration. The interface and function usage is identical to that of fnWriteXMLField as described above.
fnGetFileName	This simple function receives a full filename, which includes the file path and file extension, and returns the name of the file itself. For example, if c:\My Documents\12345.tif is passed to the function, the output is 12345. Interface is strFileName
fnGetBaseClass	This simple function returns the base class associated with the class passed to the function. If a base class is passed to the function, the same base class is returned. For example, if the function receives College and that class is a child class of Transcripts, the function returns Transcripts. Interface is strClass

Name of function / sub-routine	Description
fnIsVerifier	This function returns a Boolean value of True if the current Intelligent Capture Module executing the script is the Verifier Module.
fnGetBatchID	This function receives the path to a document in the batch directory (strWorkfile), parses the file path, and returns the batch ID number as a string. Interface is strWorkfile
fnIsAlpha	This function returns a Boolean value of True if the string passed in the strString parameter is composed entirely of alpha characters (upper or lower case). Interface is strString
fnGetUserDecimalSeparator	This function reads the local Windows settings for the user logged onto the machine and returns either a full stop/period or a comma depending on the decimal separator preferences.

fnSetDBConnection	This function can be called from a user exit to connect to a database.
	The function takes in a database connection string using the input strConnection parameter. If the connection is already available, the index of the connection in global database connection array objDBConn is returned. If it is not available or not open, the function initializes the connection and returns the relevant index of the objDBConn object.
	If the connection cannot be made, the function returns `-1` and an appropriate error message is written into the standard PIC for Transcripts log file.
	For example, the following code instantiates a database connection and execute an SQL call where variable myDBConnection represent the connection string, and mySQL represents the SQL statement (both string variables).
	`Dim lngConnection As Long`
	`Dim myConnection As ADODB.Connection`
	`lngConnection = fnSetDBConnection(myDBConnection)`
	`If lngConnection = -1 Then`
	`  ‘ Connection could not be made – error handling`
	`Else`
	`  ‘ Execute SQL using connection`
	`  Set myConnection = objDBConn(lngConnection)`
	`  myConnection.Execute(mySQL)`
	`End If`
	objDBConn is a global database object available for use in any user exit.
	Interface is strDBConnection

fnMatchDBComponents	This is a supporting function used by fnSetDBConnection.

Name of function / sub-routine	Description
fnCheckDBArray	Utility function that checks to see whether a passed database connection array of type ADODB.Connection is initialized. If it is not, the function initializes it. Interface is myArray()
fnExtractDBComponents	This is a supporting function used by fnSetDBConnection.
fnGetFieldAnalysisSettings	This function returns an instantiated AnalysisSettings object for given associative search engine field oASSA and document class strClass. Interface is strClass, oASSA
fnIsValueInList	This function takes a comma-separated list in strList input parameter and a value strValuePreserve. The function returns a Boolean true value if strValuePreserve is one of the values in the list. Interface is strList, strValuePreserve
fnConvertToDouble	This utility function takes in a string strString and converts it to a double value in a way that is consistent with the locale settings of the machine. If the string cannot be converted, the output is zero. Interface is strString
fnIsNumeric	This utility function returns a Boolean true value if all characters passed in the strTemp input parameter are numeric (For example, 0-9). Interface is strTemp
fnCheckForNull	This function receives a field component of a database record set and returns the value as a string to the calling routine using the function name. If the field component has a null value, an empty string is returned. Interface is strString
fnConvertBoolean	This function receives a Boolean field component from a database record set and returns Yes if the value is positive (and No if the value is negative) back to the calling routine using the following function name. Interface is blBool
fnSetFromFileName	This function takes the same of a parameter is the IMP section of the INI file, along with the document filename, and parses out the corresponding value from the filename passing it back to the calling routine using the function name. If the field is a date, it is formatted in accordance with the VerifierOutputFormat setting in the DAT table. Interface is strFieldName, strFile
RedimClientGlobals	This function takes the global client buffer array and initializes it if it has not already been done.

Name of function / sub-routine	Description
RedimFSGlobals	This function takes the global field settings buffer array and initializes it if it has not already been done.

fnGetClientData	This function receives a client ID and returns its corresponding settings from table Client using the ClientData structure.
	For example, to retrieve details for client zero, use the following.
	`Dim Client As ClientData`
	`Client = fnGetClientData("0")`
	Interface is strClientID

fnGetClientDataForWorkdoc	This function receives a workdoc object and returns the client settings configured in table Client that are associated with that workdoc using the ClientData structure.
	For example,
	`Dim Client As ClientData`
	`Client = fnGetClientDataForWorkdoc(pWorkdoc)`
	Interface is pWorkdoc

fnGetFieldSettings	Function to retrieve the field settings from the PICFLD table for a given combination of field name and profile ID. The settings are passed back using the FieldSettings structure.
	For example, to retrieve settings for the invoice number associated with profile ID 1, use the following.
	`Dim FS as FieldSettings`
	`FS = fnGetFieldSettings("INVOICENUMBER", "1")`
	Interface is strFieldName, strProfileID

fnReadSubRule	This function receives a substitution rule ID as a string and populates the SubRule object with the rule details, which is then passed back to the calling routine with the function name. Interface is strRule
fnGetValueForIR	This function receives the text for an invalid reason as displayed in the field in Verifier with the strIR parameter and returns a corresponding property (specified by strValue) belonging to that rule back to the calling routine with the function name. Possible values for strValue are `RULE` and `EXPORTCODE`. Interface is strValue, strIR
RedimCountryGlobals	This function takes the global country buffer array and initializes it if it has not already been done.

What is the user Exits script class?

This class contains the project user exit script points. Do not remove or change the definitions of the user exits provided. Doing so causes the solution to stop operating.

What are the Transcripts / College / HighSchool / Military script classes?

These classes contain the source code for the class validation events that includes the logic that is used to validate fields and the document as a whole, as well as to control the behavior of the Verifier form.

You can add new validation events that correspond to newly created fields on these classes. These extra events are created at the end of the existing code in the area marked in the script.

Do not make any changes to the existing code. Doing so causes the solution to stop operating.

What are the AppDev / Packaged / GenericC / GenericHS script classes?

These classes should not be deleted, changed, or renamed. Doing so causes the solution to stop operating, and the project file may not be recoverable.

What are the sequence of class dependencies?

When making changes to scripts, be mindful that dependencies exist between the various script layers, so it is not possible to execute one script if there is a dependency on a script that is not executing. Executing a script also performs a syntax check. Therefore, the scripts must be executed in the following sequence.

GlobalVariables → UserExits → Project

What are user exits?

A user exit is a dedicated public subroutine or function on the UserExits class script level where you can enter custom code.

Each user exit is called from a relevant point in the application layer baseline code and provides you with a window to perform a custom activity as is appropriate for your implementation.

You should implement customizations in a modular fashion within the existing user exits. If any ancillary functions are required to support these modules, create them as public functions. You can place these ancillary functions on the UserExits script class if they are only used locally, but if they need to be accessed by custom script on other classes, you can place them at the end of the existing script on the GlobalVariables class in the marked area.

Periodically, Perceptive Software reviews all submitted modules to ascertain which would be of value to be incorporated in the application layer baseline script.

The user exits that are available, along with their calling points and suggested uses, can be found in the following table.

User Exit	Calling routine	Description and possible uses
UserExitCustomExport	ScriptModule_ ExportDocument	This is the user exit for custom export modules, such as for custom flat files or custom database updates. This is the only user exit that has a corresponding activation parameter within the BRWEXP table. The interface is pWorkdoc, ExportPath, strDocLink, The strExportError global variable is populated with an appropriate error message in the event the export fails. This has the effect of setting the batch to a status of 750, with the error message set against the institution ID. This exit only calls for documents that have not been voided. Special handling for voided documents is inserted in the UserExitVoidDocumentExport user exit.
UserExitPostExtract	Document_ PostEvaluate on the Transcripts class	This is the user exit used to set any custom field defaults, or to reevaluate any extracted fields. The interface is pWorkdoc.
UserExitRouteDocument	ScriptModule_ RouteDocument	This is the user exit for performing any custom activity connected to the PIC for Transcripts workflow state of each document, such as changing the state based on a property of the workdoc or document filename so that they can be filtered on a user-by-user basis. The interface is pWorkdoc, State.
UserExitVoidDocument Export	ScriptModule_ ExportDocument	This is the user exit provided for the custom export of documents belonging to the void class. The interface is pWorkdoc, ExportPath, strDocLink. Populate the strExportError global variable with an appropriate error message for when the export fails. This has the effect of setting the batch to a status of 750, with the error message set against the institution ID.
UserExitTerminate	ScriptModule_ Terminate	This user exit is called from the beginning of ScriptModule_Terminate. It can be used to unload any global script objects employed in custom script. The interface is ModuleName.
UserExitPreImport	ScriptModule_ PreImport	This user exit is called from the beginning of ScriptModule_PreImport. The interface is pWorkdoc, FilePath, FileType, pCancel.

User Exit	Calling routine	Description and possible uses
UserExitPostClassify	ScriptModule_ PostClassify	This user exit is called from the beginning of ScriptModule_PostClassify. The interface is pWorkdoc.
UserExitDocument OnAction	Document_OnAction on the Transcripts class	This user exit provides an opportunity for a developer to add script that relates to custom buttons that they may elect to add to the Verifier form. The ActionName parameter, which is passed into the function, is populated with the technical name of the action associated with a user pressing the button as designated in Verifier Design Mode in the Designer Module. The interface is pWorkdoc, ActionName.
UserExitXMLOutput	Internal application	This user exit is available for a developer to add any custom fields into the XML output file. Custom fields can be entered into any of these four sections by use of the public fnWriteXMLField and fnWriteXMLDateField functions. The interface is pWorkdoc, pTable, lngLine, strSection
UserExitExportSuccess	ScriptModule_ ExportDocument	This user exit is called at the point where it is known that all selected exports have been successful for the document being processed. It can be used to update additional reporting data if required. The interface is pWorkdoc.
UserExitExportFailure	ScriptModule_ ExportDocument	This user exit is called at the point where it is known that export has failed for the document being processed. It can be used to update additional reporting data if required. The reason for the export failure can be found in the strExportError global parameter. The interface is pWorkdoc.
UserExitVerifierException	ScriptModule_Verifier Exception	This user exit is triggered when a user send a document to an exception state in Verifier. The interface is pWorkdoc, Reason, CreateNewBatch, BatchName, BatchDocumentState, BatchPriority, BatchFolderName, and ApplyExceptionHandling.

User Exit	Calling routine	Description and possible uses
UserExitSetReporting LoginName	Internal application	This user exit allows a developer to change the name of the user as reported in the Visibility reporting database. This is used in Intelligent Capture 5.2 and higher where the Web Verifier is being used. Otherwise, the system always populates the Verifier user column in the reporting database with the Intelligent Capture service user. The strUserName input parameter contains the user name that the system is currently using. The interface is pWorkdoc, strUserName. **Note** It is no longer necessary to insert code into this user exit for web verifier implementations. The system always uses the Verifier logon ID as the user name.
UserExitApplicantAddress Array	Internal application	This user exit is called each time the details for an applicant are read from the applicant pool. You can amend or add new parameters to the ApplicantAddress array. The user exit is not called if the applicant details have already been read and loaded into the local cache. The interface is oASSA, strID, ApplicantAddress
UserExitInstitutionAddres sArray	Internal application	This user exit is called each time the details for an institution are read from the institution pool. You can amend or add new parameters to the InstitutionAddress array. The user exit is not called if the institution details have already been read and loaded into the local cache. The interface is oASSA, strID, InstitutionAddress
UserExitDocument Validate	Document_Validate on the Invoices class script level	This user exit is called from Document_Validate on the Invoices class script level. It can be used to code in additional document level validations and activities. The interface is pWorkdoc, pValid.
UserExitEditDocument Weblink	Internal application	This user exit permits a developer to manipulate the document web link, as stored in the Visibility reporting database and exported downstream. The current web link is passed into the user exit using the strWebLink interface parameter, and this may be changed to meet your business needs. The current unique document ID is passed in the strDocID interface parameter, and this cannot be changed. The interface is strWebLink, strDocID.
UserExitVerifierFormLoad	ScriptModule_Verifier FormLoad	This user exit is called at the end of ScriptModule_VerifierFormLoad. The interface is pWorkdoc, FormClassName, FormName

User Exit	Calling routine	Description and possible uses
UserExitScriptModule Initialize	ScriptModule_ Initialize	This user exit is called at the end of ScriptModule_Initialize. The interface is pWorkdoc.
UserExitPostImport	ScriptModule_ PostImport	This user exit is called from the beginning of ScriptModule_PostImport. The interface is pWorkdoc.
UserExitPostImportBatch	ScriptModule_Post ImportBatch	This user exit is called at the beginning of ScriptModule_PostImportBatch. The interface is pWorkdoc.
UserExitPreClassify	ScriptModule_Pre Classify	This user exit is called from the beginning of ScriptModule_PreClassify. Interface is pWorkdoc

What are custom error messages?

If script code placed within the user exit framework is to include custom error messages, then these may be included as entries in the BRWERR table rather than being hard coded within the script. The error message number range is 900-999, which should not be modified to prevent any conflicts in the event of an upgrade. For example, in the BRWERR table, the following row has been added.

ErrorNumber	Message
900	Please check data entry.

This can be retrieved through the script with the following line of code.

```
Dim myError As String
myError = DicVal(“900”, “ERR”)
```

The local strError string now contains the following.

```
Please check data entry
```

What are the project data structures?

PIC for Transcripts uses internal data structures to pass data between functions and subroutines. It is possible to use some of these data structures in user exit scripts. In some cases, these structures are defined as formal parameters in the interface. The following structures are available.

- ApplicantAddress
- InstitutionAddress
- ClientData
- FieldSettings

What is the ApplicantAddress structure?

The ApplicantAddress structure contains data elements associated with a particular applicant, such as the applicant ID, the applicant name, address details, and additional information. The extent to which the data is populated depends on the extent to which the data is available in the applicant extract and mapped within the BRWSRA table. It is used in the interfaces to the UserExitApplicantAddressArray user exit.

The structure consists of the following elements.

Structure element	Type	Description
FNAME	String	This is the first name of the applicant.
MNAME	String	This is the middle name of the applicant.
LNAME	String	This is the last name of the applicant.
ADDRESS	String	This is the applicant street address line 1.
ADDRESS2	String	This is the applicant street address line 2.
ZIP	String	This is the applicant zip / postal code.
ID	String	This is the unique applicant ID from the point of view of the data extract where each row must have a unique reference. This is not the unique applicant ID from the point of view of the SIS if a site ID is also used.
SITEID	String	This is the applicant site ID.
TELNo	String	This is the applicant telephone number.
CITY	String	This is the applicant city.
STATE	String	This is the applicant state. For US addresses, the state code is expected here. For example, CA=California, VA=Virginia, and so on.
DOB	String	This is the applicant date of birth.

Structure element	Type	Description
SSN	String	This is the applicant social security number.
APPLICANTIDENTIFIER	String	This is the unique applicant identifier code. For example, SSN.
PARTITIONID	String	This is the applicant partition ID.
EXTERNALID	String	This is the SIS system applicant ID if a site ID is being used.
CUSTOM1	String	This is the a custom value that allows the export of additional data with each applicant.
CUSTOM2	String	This is the a custom value that allows the export of additional data with each applicant.
CUSTOM3	String	This is the a custom value that allows the export of additional data with each applicant.
CUSTOM4	String	This is the a custom value that allows the export of additional data with each applicant.
CUSTOM5	String	This is the a custom value that allows the export of additional data with each applicant.

What is the InstitutionAddress structure?

The InstitutionAddress structure contains data elements associated with a particular institution, such as the institution ID, the institution name, address details, and additional information. The extent to which the data is populated depends on the extent to which the data is available in the institution extract and mapped within the BRWSRC table. It is used in the interfaces to the UserExitInstitutionAddressArray user exit.

The structure consists of the following elements.

Structure element	Type	Description
NAME	String	This is the name of the institution.
ADDRESS	String	This is the institution street address line 1.
ADDRESS2	String	This is the institution street address line 2.
ZIP	String	This is the institution zip/postal code.
ID	String	This is the unique institution ID from the point of view of the data extract where each row must have a unique reference. This is not the unique institution ID from the point of view of the SIS if a site ID is also used.
SITEID	String	This is the institution site ID.

Structure element	Type	Description
TELNo	String	This is the institution telephone number.
CITY	String	This is the institution city.
STATE	String	This is the institution state. For US addresses, the state code is expected here. For example, CA = California, VA = Virginia.
COUNTRY	String	This is the institution country This is the two-character ISO code for the country. For example, US = United States Of America, DE = Germany.
CEEB	String	This is the institution CEED code.
ACT	String	This is the institution ACT code number.
FICE	String	This is the institution FICE.
OPEID	String	This is the institution OPE ID.
INSTITUTIONIDENTIFIER	String	Unique institution identifier code.
PARTITIONID	String	This is the institution partition ID.
EXTERNALID	String	This is the SIS system institution ID if a site ID is being used.

What is the ClientData structure?

The ClientData data structure is used to hold details of the current client. It can be read from any routine where pWorkdoc is available with the use of the fnGetClientDataForWorkdoc global function.

The structure consists of the following elements.

Structure element	Type	Description
CLIENTID	String	This is the client ID.
PROFILEID	String	This is the profile ID assigned to client
EXPORTPROFILEID	String	This is the export profile ID assigned to client.
FORCEVERIFY	Boolean	This is the flag denoting whether all fields require verification.
CLIENTGROUP	String	This is the verifier access ID assigned to a client.
CLIENTNAME	String	This is the client name.
INSTRUCTIONS PROFILEID	String	This is the instructions profile ID assigned to client.

Structure element	Type	Description
REQUIRESREVIEW	Boolean	This is the requires review flag.
INSTITUTIONPARTITION	String	This is the institution partition ID assigned to client.
APPLICANTPARTITION	String	This is the applicant partition ID assigned to client.
PRIORITY	String	This is the batch priority level for client.

What is the FieldSettings structure?

The FieldSettings structure holds details associated with a given field as read from the BRWFLD table. The structure consists of the following elements.

Structure element	Type	Description
FIELDNAME	String	This is the field name.
PROFILEID	String	This is the profile ID.
VERIFIERLABEL	String	This is the field verifier label.
ACTIVE	Boolean	This is the field active flag.
REQUIREDINRTS	Boolean	This is the denotes whether the field is required in RTS.
REQUIREDINVERIFIER	Boolean	This is the denotes whether field entry is mandatory in Verifier.
COUNTRYFILTER	String	This is the comma-separated list of countries that control whether the field is mandatory or not.
FIELDTYPE	String	This is the field type.
FORCEVERIFY	Boolean	This is the force verify indicator.
DEFAULTVALUE	String	This is the field default value.
DEFAULTIFNoTHINGEXTR	String	This is the field default if no value is extracted automatically.
SUBRULE	String	This is the field substitution rule.
MINLENGTH	Integer	This is the field minimum length.
MAXLENGTH	Integer	This is the field maximum length.
RIGHTJUSTIFY	Boolean	This is the indicator as to whether the field is right justified if a pad character is used.
PADCHAR	String	This is the padding character.

Structure element	Type	Description
REMOVEALLSPECIALS	Boolean	This indicates whether special characters are removed.
REMOVEBLANKS	Boolean	This indicates whether blank spaces are removed.
KEEPCERTAINSPECIALS	String	This is the list of special characters that are retained.
REMOVESTARTEND	Boolean	This is the indicates whether special characters is removed from the start and end of the string.
SUBSTRINGSTARTPOS	Integer	This is the substring start position.
SUBSTRINGLENGTH	Integer	This is the substring length.
REMOVELEADINGZEROS	Boolean	This is the flag to indicate whether leading zeroes is removed from a string.
DECIMALPLACES	Integer	This is the number of decimal places for an exported amount.
NEGATIVETYPE	Integer	This is the negative type code.
OUTPUTFORZERO	String	This is the export value if an amount field is zero.
SUBSTITUTEVALUEIF OVER0	String	This is the export value if an amount is greater than zero.
FUTUREDAYS	Long	This is the number of days that an extracted date is permitted to be in the future.
NoDAYSINPAST	Long	This is the number of days that an extracted date is permitted to be in the past.
DATEONLYINCURRENT MONTH	Boolean	This is the indicates whether the date should only be in the current month.
FIELDMASK	String	This is the list of valid field masks for text fields.

About the triggering of user exits in Verifier

User exits are triggered when a user is working a problem document in Verifier.

The following table lists the user exits that are fired when a user performs a certain task in the order in which they are fired.

Verifier action	User exits
A user clicks on a button on the Verifier form.	UserExitDocumentOnAction
A user verifies the last invalid field on the Verifier form.	**Always calls** UserExitDocumentValidate **Potentially calls** UserExitSetReportingLoginName if Visibility reporting is activated.

Add the report and custom base classes

If the project involves adding a new base class, then the standard reporting audit trail is not complete without the following steps being performed. The steps are as follows.

1. Add script into Document_PreExtract and Document_Validate events on the custom base class.
2. Create the custom tmpCLSRES field on the base class.

Add a custom script to the Document_PreExtract and Document_Validate events

To insert a custom script into the Document_PreExtract and Document_Validate events, complete the following steps.

1. Open **Intelligent Capture Designer**.
2. In **Definition** mode, in the **Class** view, highlight **CustomBaseClass**, then right-click and select **Show script**.

3. Copy the following script into the script window.

```
Private Sub Document_PreExtract(pWorkdoc As SCBCdrPROJLib.SCBCdrWorkdoc)
  If InStr(UCase(ScriptModule.ModuleName), cVerifier) Then
    gblVerifierAsServer = True
  Else
    gblVerifierAsServer = False
  End If
  fnGetClassResultsMatrix(pWorkdoc)
  fnReporting(pWorkdoc, "DOCUMENTPREEXTRACT")
End Sub
Private Sub Document_Validate(pWorkdoc As SCBCdrPROJLib.SCBCdrWorkdoc, pValid As
Boolean)
  gblVerifierAsServer = False
  If UCase(ScriptModule.ModuleName) <> cVerifier Then
    fnReporting(pWorkdoc, "DOCUMENTVALIDATESERVER")
  Else
    fnReporting(pWorkdoc, "DOCUMENTVALIDATEVERIFIER")
  End If
End Sub
```

What is the tmpCLSRES field?

The tmpCLSRES field is used to store the full classification results and weightings before they are written to the reporting database. It is an internal field and does not require any action beyond its creation. If the field is not created, the full classification results and weightings are not written into the reporting database, only the final class in which the document was placed.

Create the tmpCLSRES field

To create the tmpCLSRES field, complete the following steps.

1. In the custom base class, go to the Fields view mode.
2. Right-click in the grey space and select **Insert Field Definition**.
3. Enter `tmpCLSRES`. This is case sensitive.
4. Right-click on the new field and select **Show Properties**.
5. In the pane on the right side, select the **Validation** tab, and then select the **Always Valid** flag.
6. Save the project.

About configuring data exports

The following data exports are available in PIC for Transcripts.

- Export an additional TIFF image.
- Export a PDF file.
- Write data to an XML file.
- Set up a custom export.

Data export is controlled by settings in the following tables. Exports only apply to documents classified to the Transcripts class or to one of its child classes. For custom base classes, the data export needs to be coded programmatically within UserExitCustomExport on the UserExits script class level. Each of the tables is keyed by an export profile ID that can be assigned to a client.

- BRWEXP
- BRWEXPHSHeader
- BRWEXPMilCourses
- BRWEXPStudentHeader
- BRWEXPUnivCourses
- BRWEXPUnivDegree
- BRWEXPUnivRecord

One TIFF image is always output to the export directory when a transcript is added to the system. It is recommended that this setting should always be used over custom settings. However, custom settings provide additional options for outputting a second TIFF image and are used when one or more of the following situations apply.

- A second TIFF file is required during document export.
- The TIFF file name must be set to the document URN rather than the original image file name.
- The TIFF image resolution needs to be changed from the original image.
- The TIFF image compression ratio needs to be changed from that of the original image.

About exporting an additional TIFF image

The additional TIFF file is always written to the directory specified as the export directory on the runtime server instance settings for the instance that is carrying out the document export. If you do not specify an export directory, the default export directory in the DefaultExportPath setting is used. If the setting is left blank, the document export fails and the batch goes to a status of 750.

Export an additional TIFF image

To export an additional TIFF image, complete the following steps.

1. Open the **BRWEXP** table.
2. Set the **OutputTiffFile** column to `True`.
3. To set the name of the TIFF file to match the document URN in the *<project>*.ini file, set the **TiffName** column to `URN`. Otherwise, the name is the same as the original imported document.

Configure the DPI and image compression type

Configuring the DPI and image compression types is optional. To change the image DPI and compression, complete the following steps.

1. In the **BRWEXP** table, set **TiffDPI** column to a DPI of your choice. The default image compression is 300 dots-per-inch.
2. In the same column, set image compression to one of the following. The default is standard Grade 4 compression.
   - **G4FAX** This is standard Grade 4 compression.
   - **G3FAX** This is standard Grade 3 compression.
   - **LZWFAX** This is LZW compression.
   - **HUFFAX** This is HUF compression.

About exporting a PDF file

You can output a transcript as a searchable PDF file. The PDF file is always written to the directory specified as the export directory on the runtime server instance settings for the instance that is carrying out the document export. If you do not specify an export directory, the default directory in the DefaultExportPath setting is used, the export fails, and the batch goes to status 750.

Export a PDF file

To export a PDF file, complete the following steps.

1. Open the **BRWEXP** table.
2. Set the **OutputPDF** column to `True`.
3. To set the name of the PDF file to match the document URN in the *<project>*.ini file, set the **PDFName** column to `URN`. Otherwise, the name is the same as the original imported document.

About configuring the XML file

The following sections contain the steps to configure the export of data to an XML file, and how the XML file can be set up to include custom fields. The XML file itself, and the following output instructions, are divided into separate sections based on the document type.

What are the XML file sections?

The standard XML output file is divided into separate sections. Some sections are the same regardless of the document type, while other sections are dependent on the document type. The following sections are the same for all document types.

- **Document section** This section includes global document fields such as the file name, the document type, and the date issued.
- **Student record section** This section includes fields associated with the applicant such as the applicant's ID, first name, and last name.

XML file sections for college transcripts

The following sections of the XML file are reserved for college transcripts.

University degree This section includes information such as the degree earned and graduation date.

University institution This section includes information pertaining to the issuing institution, such as the institution ID, institution name, FICE and OPE ID.

University course records This section contains the transcript coursework information, written line-by-line, grouped by Term, and includes information such as the course number, course description, and grade.

XML file sections for high school transcripts

The following sections of the XML file are reserved for high school transcripts.

High school header This section Includes header fields that are extracted from high school transcripts, such as cumulative GPA, weighted GPA and total credit hours.

High school institution section Includes information pertaining to the issuing institution such as institution ID, institution name, FICE and OPE ID.

XML file section for military transcripts

The following section of the XML file is reserved for military transcripts.

Military course records The transcript coursework information in this section is written in line-by-line and includes information such as the ACE identifier, course number and course title.

Activate output to an XML file

To set up output to an XML file, complete the following steps.

1. Open the ***<project>*.ini** file.
2. To set the name of the XML file to match the document URN, in the ***<project>*.ini** file, set the **XMLFileName** column to `URN`. Otherwise, the name is the same as the original imported document.
3. Open the **BRWEXP** table.
4. In the **DefaultExportPath** column, enter the path to the output directory.

5. Optional. To set the file extension to .xml, set the **XMLFileType** column to `True`. The default extension is .xml.
6. Optional. To support UTF-16 for non-western characters, such as Chinese, set the **XMLEncodingHeader** column to `UTF-16`.

   Note An XML file is not generated if the output data contains non-western characters and the encoding header is set to UTF-8.

Additional XML file section options

Configure the following document fields and section tags in the **BRWEXP** table.

BRWEXP Column	Description
XMLFileHeader	This is the document header tag.
XMLFileHeaderAttributes	Optional. This controls the attributes assigned to the document header tag, such as namespace information.
XMLDocName	This is the exported document file name.
XMLType	This denotes the transcript type, college, high school or military.
XMLIssueDate	This is the date the transcript was issued.
XMLStudentHeader	This is the student record section.
XMLUniversityHeader	This is the university degree section.
XMLUniversityHeader	This is the university degree section.
XMLUniversityDegreeHeader	This is the university degree subsection.
XMLUnivRecordHeader	This is the university institution section.
XMLHighSchoolRecordHeader	This is the high school institution section.
XMLHighSchoolHeader	This is the high school header section.
XMLUnivCourseHeader	This is the university course records section.
XMLUnivCourseTag	This is the each university.
XMLMilCourseHeader	This is the military course records section.

About defining field output

After the XML output has been activated and you have configured the section tags, you configure the fields that are written into the file and define how they are tagged. The field output is configured in the following tables. Tags for each field and section are defined in the XML file sections in the corresponding tables.

Configuration Table Name	XML Section
BRWEXPHSHeader	High School Header
BRWEXPMilCourses	Military Course Records
BRWEXPStudentHeader	Student Record
BRWEXPUnivCourses	University Course Records
BRWEXPUnivDegree	University Degree
BRWEXPUnivRecord	University and High School Institution

Define field output

To configure a field in the XML file, complete the following steps.

1. Populate the column **XMLTag** with the desired XML tag value for the given field.
2. Repeat the previous step for all fields in the XML section.

For example, if the Cumulative GPA is to be written into the XML file with a tag of CUMGPA, the row representing the Cumulative GPA field should have CUMGPA populated in the XMLTag column.

Note If there is a field that you do not want written into the XML file, leave the XMLTag column for the field row blank.

About setting up custom exports

If you have a required data export and the existing export options do not support the data export's format, or if you need to export data for a custom base class, you must create a custom export. The custom export must be scripted and executed within a special user exit. The following sections describe how to implement a custom export.

The user exit is called once for each document that is exported. Once a document is exported, the export history is updated against the document so that it is not unintentionally exported a second time. The history can be cleared by resetting the document back to state 200. If an export is not successful, the user exit is called again during the next attempt.

The script contents of a user exit can be set to anything that your business needs require.

You should check the document class before developing any script that refers to fields using hard-coded field names, particularly if the project uses custom base classes. If a field is that does not exist is referenced against the document class, it results in a runtime error. The fnGetBaseClass global function, described in the Global Variables Scrip Class, can be used to check the document class.

Configure and activate a custom export

To configure and activate a custom export, complete the following steps.

1. Open the project file with the **Designer** module.
2. Navigate to **Definition** mode and highlight the **UserExits** class.
3. Right-click on the class and select **Show Script**.
4. Navigate to the **UserExitCustomExport** user exit subroutine.
5. Configure the parameters listed in the following table.
6. Activate the custom export by setting the **CustomExport** column in the **BRWEXP** table to `True`.

Parameter name	Description
pWorkdoc	This is the standard Workdoc object that provides access to all document field information, including the originally extracted line item data, the document classname, the document OCR text, and the document file name.
ExportPath	This is the destination folder for file output. This value is taken from the export file path configured on the RTS instance that is responsible for document export. If the RTS instance path is blank, the export path is set to the value held against the system configuration EXP_VL_DefaultExportPath parameter.
strDocLink	This is the path to the image of the document, which can be stored in a storage director or the batch directory, or as a URL, to retrieve the image from an archive.

About adding and configuring clients

PIC for Transcripts is a solution that permits multiple configuration types to operate within a single installation. A single configuration type is referred to as a client. Each document that passes through the system is preassigned to a client, and the client controls the following processes.

- The overall document flow.
- The fields that are extracted.
- Those fields that are mandatory and those that are optional, and their corresponding validation rules.
- The data sources that are used for field validation.
- How data is exported.

When you are designing a PIC for Transcripts client, consideration is given to how the client is utilized for your business needs, for example.

- If the end user is a BPO, a client can be used to represent a single customer of the BPO or a division of a single customer.
- If you have one user working in multiple regions or with multiple divisions with their own requirements, a client can be used to represent each region or division.
- If you have one user working with multiple SIS systems, each SIS system can be set up as an individual client for the different SIS-system connections and processing rules.

What are the client settings and properties?

Client settings and properties are contained in the BRWClient table in the PIC for Transcripts database. A basic installation creates a single client with a client ID of 0 (zero), and this is the default client the system uses.

The columns contained in the BRWClient table and their uses are described in the following table.

Column	Explanation of usage
ClientID	This is the unique ID of the client and must always be set to an integer value.
ProfileID	This is ID of the profile assigned to the client. The profile controls what fields are extracted and how they are validated. More than one client may share the same profile ID if the extraction and validation requirements are identical.
ExportProfileID	This is the ID of the export profile assigned to the client. The export profile ID controls how data is exported for a client. More than one client may share the same export profile ID if the export requirements are identical.
ClientName	This is a free text string containing the name of the client. This data is written to the Visibility reporting database for each document assigned to a client.
ForceVerify	This is a flag that controls whether all documents for a client is routed to the Dynamic Verifier. If this column is set to True, all documents are routed to the verifier. If it is set to False, only documents requiring review by a user attention are routed to the verifier.
ClientGroup	This is the ID of the client group to which the client belongs. It is an integer value that can be set by the system administrator. You use this group to give users access to documents belonging to specific clients.
RequiresReview	This is a flag that indicates whether documents assigned to a client should always be subject to review by a user after the document has been routed through the Dynamic Verifier.
InstitutionPartition	This is the ID of the institution master data partition that is used by the client.
ApplicantPartition	This is the ID of the applicant master data partition that is used by the client.
Priority	When documents are imported into Intelligent Capture, they are placed in batches and each batch is assigned a priority. This priority controls the order by which the runtime server component of Intelligent Capture processes the batches, and the order in which the documents appear in the Dynamic Verifier. The priority scale runs from 1 to 9, and 1 is the highest level of priority. If you set this to 1, all batches from this client have a priority of 1.

About assigning documents to a client

Documents must be preassigned to clients prior to being captured by Intelligent Capture. Intelligent Capture uses a parameter in the image file name to identify the client a document is assigned to. Therefore, a client ID must be embedded within the image file name and separated by an underscore. The part of the file name that represents the client ID is specified in the IMP section in the <*project*>.ini file in the IMP_VL_ClientID parameter with the word COMPONENT followed by a number that indicates the client ID's position in the file name.

For example, processing a document using the component assigned to client 2 has a client ID embedded in the file name as follows.

```
12345_2_20120901.tif
```

The first component of the file name is 12345, the second component and client identifier is 2, and the third component is 20120901. The <*project*>.ini file setting is configured as follows.

```
IMP_VL_ClientID=COMPONENT2
```

If the file name for client 2 is 12345_20120901_2.tif, the <*project*>.ini file setting is configured as follows because the client identifier is located in the file name's third component.

```
IMP_VL_ClientID=COMPONENT3
```

Note If this parameter is not set or the file component does not exist, the system processes the document using the default configuration assigned to client zero.

About creating institution and applicant master partitions

The choice to implement institution and applicant partitions must be made during the installation and setup process.

PIC for Transcripts supports multiple sets of institution and applicant data within the same project file. Each set of master data is referred to as a partition and is assigned its own partition ID within the system. Partitions IDs are, in turn, assigned to clients in the BRWClient table. Multiple clients may share the same partitions.

When the institution and applicant are being determined by the system at runtime, the system only takes into account institutions and applicants that belong to the institution and applicant partition assigned to the client. Within the Dynamic Verifier application, when the user executes a search, only institutions and applicants assigned to that client are included in the results.

About implementing a partition

The following list provides a high-level overview of the steps for implementing a partition.

1. Activate partitioning within the <*project*>.ini file.
2. Register the institution and applicant partitions in the BRWINSPartition and BRWAPTPartition tables.
3. Assign the partition IDs to the client.
4. Populate the institution and applicant master tables.
5. Create a user DSN for the institution and applicant master tables.
6. Configure the ASA sections in the <*project*>.ini file.
7. Create the ASE pools.
8. Configure the BRWSRC (institution) and BRWSRA (applicant) tables.

Activate partitioning

To activate partitioning, complete the following steps.

1. Open the **<*project*>.ini** file.
2. Navigate to the **Global** directory.
3. Set the following parameters.

```
GRL_OP_ActivateInstitutionFiltering=Yes
GRL_VL_InstitutionFilterColumn=PartitionID

GRL_OP_ActivateApplicantFiltering=Yes
GRL_VL_ApplicantFilterColumn=PartitionID
```

4. Save and close the INI file.

Register the partition

To register the partitions, complete the following steps.

1. Open the **BRWINSPartition** and **BRWAPTPartition** tables.
2. Populate a row with a partition ID. This must be an integer.
3. Add a description. Adding a description of the partition is optional, but its naming is indicative of what it represents.
4. Save the changes.

Assign the partition ID to the client

To assign the partition ID to the client, complete the following steps.

1. Open the **BRWClient** table.
2. In the corresponding **InstitutionPartition** and **ApplicantPartition** columns, enter the ID of the newly registered partition for the appropriate client.
3. Save the changes.

Populate the master tables

Included within the PIC for Transcripts database are sample BRWInstitutionMaster, BRWApplicantMaster institution, and applicant master tables. These contain an example structure that the tables should follow. To populate the master tables, complete the following steps.

1. Open the BRWInstitutionMaster and BRWApplicantMaster master tables.
2. Populate the following columns in the tables.
3. Optional. Populate additional columns according to your business needs.
4. Save the changes.

Column	Explanation of usage
Index	This is a unique identifier for the record in the table. This value is set to the partition ID followed by a hyphen, and then the client's institution or applicant ID.
PartitionID	This is the partition ID for the institution or applicant master as set in the BRWClient, BRWINSPartition, and BRWAPTPartition tables.
ID	This is the client's institution or applicant ID.
Address1	This is the street address of the institution or applicant.
City	This is the city of the institution or applicant.
Zip	This is the zip code of the institution or applicant.

Create a user DSN for the institution or applicant master tables

Intelligent Capture requires a user DSN to be created which reflects a connection to the PIC for Transcripts database using SQL Server-based authentication. You create the user DSN using Administrative Tools on a Windows machine. For more information about creating a user DSN, refer to Windows documentation.

Configure the ASA section in the INI file

To configure the ASA section in the INI file, complete the following steps.

1. Open the **<*project*>.ini** file.
2. In the **ASA** section, configure the following settings, replacing `myDSN` with the name of the user DSN created in the previous section, and `myUSERNAME` and `myPASSWORD` with the appropriate database credentials.

```
ASA_VL_01_Class=Transcripts
ASA_VL_01_Fieldname=InstitutionASE
ASA_OP_01_AlphaNum=Yes
ASA_OP_01_PoolRelative=Yes
ASA_VL_01_PoolPath=
ASA_VL_01_PoolDirectory=Pool
ASA_VL_01_PoolName=Institutions
```

```
ASA_OP_01_FileRelative=Yes
ASA_VL_01_ImportPathFilename=
ASA_VL_01_ImportFilename=
ASA_VL_01_ImportODBCDSN=myDSN
ASA_VL_01_ImportODBCSelect=select * from BRWInstitutionMaster
ASA_VL_01_ImportODBCUser=myUSERNAME
ASA_VL_01_ImportODBCPWD=myPASSWORD
ASA_VL_01_AutoImportOption=ODBC
```

3. Save and close the INI file.

Create the ASE pool

To create the ASE pool, complete the following steps.

1. Open the **<*project*>.ini** file and navigate to the **InstitutionASE/ApplicantASE** on the **Transcripts** class.
2. Display the field settings.
3. On the **File Import** tab, to import the pool from the database table, click **Import**. A message stating that the pool was created is displayed. If an error message is displayed, the institution or applicant master tables have not been configured correctly. Correct any configuration errors and reimport the pool.
4. On the **General** tab, complete the following substeps.
    1. In the **ID** column, set the radio button to the field that is the unique identifier for the institution or applicant row in the database.
    2. In the **Filter** column, set the radio button to the **PartitionID** field.
    3. In the **Search** column, only select those field values that are strong and unique criteria for selecting the institution or applicant.
5. On the **File import** tab, complete the following substeps.
    1. Reimport the **pool**.
    2. Set the class settings to the following configuration: `[Name]_[Index]`
    3. Configure the following field settings, set the first line to the unique identifier for the record in the institution or applicant extract. It is recommended that the field uses the following structure, but this is optional depending on what is appropriate for the client as long as the first line is the unique identifier.

        ```
        [Index]
        [Name]
        [Address1]
        [City]
        [State] [Zip]
        ```

6. The institution or applicant field configuration is complete, and a green light with the message **Engine Is Ready** should appear in the field status box. Save and close the project file.

Configure the BRWSRC and BRWSRA tables

The BRWSRC and BRWSRA tables tell the system which column in the institution master or applicant master pool corresponds to the internal field that is used during processing. Both are global tables and should only have a single row. To configure the tables, complete the following step.

- Map the column in the **BRWSRC** and **BRWSRA** tables to the column name of the database column within the master table or the CSV file column header for the respective ASE pool.

Working without institution or applicant partitions

If the project does not require an institution or applicant partition, because it is for a single client or multiple clients that pool the same set of institution or applicant data, complete the following step to set up your environment without partitions.

- In the ***<project>*.ini** file, set the following parameters.

  ```
  GRL_OP_ActivateInstitutionFiltering=No
  GRL_OP_ActivateApplicantFiltering=No
  ```

What are field configuration?

The following sections contain the instruction for configuring fields. The extraction and validation of fields is controlled in the PIC for Transcripts database in the BRWFLD table. Extraction and validation rules are set at the profile ID level and assigned to clients.

Each row in the database represents a field, and the table is keyed by the profile ID and the technical name of the field. During installation, the table is populated with a full list of the fields available within the project for Client 0 (zero).

The entries in BRWFLD table allow you to do the following.

- Switch fields on and off.
- Set fields to mandatory or optional.
- Set default field values.
- Set a field type, such as date, amount, table, text, and corresponding validation rules.

Note The name of the field is displayed in the FieldName column. This name should not be changed.

About switching fields on and off

Fields that are not active do not appear on the Dynamic Verifier form. If a standard field in the project is not listed in BRWFLD table for a profile, it is considered inactive. The following fields in the Coursework table cannot be turned off.

- InstitutionID
- InstitutionASE
- ApplicantID
- ApplicantASE
- TranscriptType
- InvalidReason

Switch fields on and off

To turn a field on or off, complete the following steps.

1. Open the **BRWFLD** table.
2. To turn a field on, set the **Active** column to `True`.
3. To turn a field off, set the **Active** column to `False`.
4. Save the changes.

Set a field to be mandatory or optional

Whether a field is mandatory or optional is controlled using the RequiredInRTS and RequiredInVerifier columns in the BRWFLD table. The following describes the effect of setting these columns to True and False in isolation and in tandem.

RequiredInRTS	RequiredInVerifier	Effect
False	False	Population of the field is optional within the project.
True	False	The field is marked invalid and the document sent to Verifier if the system does not extract a value into this field automatically. The user is permitted to pass a blank value in the Verifier application.
True	True	The field is marked invalid and the document sent to Verifier if the system does not extract a value into this field automatically. The user must enter a value in Verifier.
False	True	The field is marked invalid and the document sent to Verifier if the system does not extract a value into this field automatically. The user must enter a value in Verifier. This exhibits the same behavior as the True/True option.

Set a field to be mandatory or optional

To make a field mandatory or optional, complete the following steps.

1. Open the **BRWFLD** table.
2. To make a field mandatory, set the **RequiredInRTS** and **RequiredInVerifier** columns to `True`.
3. To make a field optional, set the **RequiredInRTS** and **RequiredInVerifier** columns to `False`.
4. Save the changes.

Force a field to appear in Verifier

To configure a field so it is always marked invalid and then reviewed by a user in Verifier, complete the following steps.

1. Open the **BRWFLD** table.
2. Set the **ForceVerify** column to `True`.
3. Save the changes.

Label a field in Verifier

To control how a field is labeled in Verifier, complete the following steps.

1. Open the **BRWFLD** table.
2. In the **VerifierLabe**l column, enter the text that you want to appear in the Dynamic Verifier.
3. Save the change.

Set field default values

There are two default settings for every field, and the field usage depends upon how the default is applied. To set the default values for a field, complete one of the following steps.

- If a field should always be set to a fixed value irrespective of extraction, set the **DefaultValue** column to `True`.
- If a field should default to a value because the system has not extracted anything into that field, set the **DefaultIfNothingExtr** column to `True`.

About configuring field types

The following are the four field types you can assign to each field.

- Date
- Amount
- Text
- Table

The field type governs which of the additional settings in the table affect the validation of the field. The different types, along with their configurations, are described in the following sections.

Configure date fields

To configure date fields, complete the following steps.

1. Open the **BRWFLD** table.
2. In the **FutureDays** column, enter a numerical value that indicates the number of days in the future from the present date that an extracted date is considered valid. If future dates are not permitted, then this is set to `0` (zero).
3. In the **NoDaysInPast** column, enter a numerical value that indicates the number of days in the past from the present date that an extracted date is considered valid. If future dates are not permitted, then this is set to `0` (zero).
4. To force an extracted date to stop in Dynamic Verifier if the displayed date is not in the current month, set the **DateOnlyInCurrentMonth** column to `True`.
5. To have the date in the Dynamic Verifier displayed as MM/DD/YYYY, in the **VerifierOutputFormat** column, enter `MMDDYYYY`, or to display the date as DD/MM/YYYY, enter `DDMMYYYY`.

   Note The date output format settings are set for each profile ID, so a different configuration is permitted for each client.
6. Save the changes.

Configure amount fields

To configure amount fields, complete the following steps.

1. Open the **BRWFLD** table.
2. Set the **Amount** field to `Amount`.
3. To control the number of decimal places assigned to the amount during data export, in the **DecimalPlaces** column, enter a numeric value.
4. To set the output of an extracted value that is less than zero, in the **NegativeType** column, enter one of the following integers.
   - `1` This forces a minus sign to appear after an amount, such as 100.00-.
   - `2` This forces a minus sign to appear before an amount, such as -100.00.
   - `3` This forces a value to appear in parentheses, such as (100.00).
5. If an alternative value is displayed if the extracted value is either empty or zero, in the **OutputForZero** column, enter the alternate value.
6. If an alternative value is displayed if the extracted value is greater than zero, in the **OutputForZero** column, enter the alternate value.

Format amounts during output

The BRWAMT table contains two settings that control the formatting separators for amounts during output. The following table shows the effect upon the output of the amount 10,000 using a combination of the settings if the number of decimal places is set to 2.

To configure separators for amounts during output, complete the following steps.

1. Open the **BRWFLD** table.
2. In the **ExportThousandSeparator** column, enter a separator type.
3. In the **ExportDecimalSeparator** column, enter the second separator type.
4. Save the changes.

ExportThousandSeparator	ExportDecimalSeparator	Output for a value of 10,000 to 2 decimal places
,	.	10,000.00
.	,	10.000,00
(blank)	.	10000.00
(blank)	(blank)	10000.00
‘	.	10’000.00

Configure text fields

Text fields are fields that can contain numeric and alphanumeric characters. To configure text fields, complete the following steps.

1. Open the **BRWFLD** table.
2. In the **MinLength** column, enter a numeric value that represents the minimum permitted length of the field.
3. In the **MaxLength** column, enter a numeric value that represents the maximum permitted length of the field.
4. In the **PadChar** column, enter a numeric value that is the length a field is padded to the right when a maximum field length is not met, such as 123400000.
5. To have a value padded by the character entered on the left, such as 000001234, set the **RightJustify** column to `True`.
6. To remove all special characters from a value, set the **RemoveAllSpecials** column to `True`.
7. To remove all spaces from a value, set the **RemoveBlanks** column to `True`.
8. To retain non-comma separated lists of special characters that are retained when the RemoveAllSpecials column is set to true, set the **KeepCertainSpecials** column to `True`.
9. To have positive numbers start from the left side of a field and negative numbers start from the right side, set the **SubstringStartPos** column to `True`.
10. To remove any leading zeroes from an extracted or user-entered value, set the **RemoveLeadingZeros** column to `True`.
11. To enable a field to have a comma separated list of valid entries that contains only approved characters for an extracted or user entered value, in the **FieldMask** column, enter any letter, number or wildcard character. For example, if the content of this column is set to `ABCD,WXYZ` then no value is permitted in this field unless it is equal to either ABCD or WXYZ.

About using substitution rules

In the BRWFLD table, a text field can be assigned a substitution rule, which permits an extracted text value to be substituted in part, or as a whole, with another value. This works in a similar way to the standard VB replace command. One substitution rule can be assigned per field.

Substitution rules are contained in the BRWSubstitution global table, which has the following structure.

Fieldname	Explanation of usage
Index	Substitution rule index.
Original	String value to be replaced.
Replace	String value to be substituted.

What are Verifier forms?

A Dynamic Verifier form is a window presented to users in the Verifier application that contains fields that are active and that can be updated by users. A Dynamic Verifier form is activated in the *<project>*.ini file. When a Dynamic Verifier form is activated, all documents imported into Intelligent Capture use the Dynamic Verifier form from that point forward. The following sections contain the instructions for configuring and activating Dynamic Verifier forms.

The following fields cannot be deactivated and appear in a Dynamic Verifier form even though they cannot be updated by users.

- InstitutionID
- InstitutionASE
- ApplicantID
- ApplicantASE
- TranscriptType
- InvalidReason

Activate a Dynamic Verifier form

To activate a Dynamic Verifier form, complete the following steps.

1. Open the ***<project>*.ini** file.
2. Set `GRL_OP_UseDynamicVerifierForm` to `Yes`.
3. Save the changes.
4. Restart the **Verifier** application on each client machine.

Turn off Dynamic Verifier logging

Diagnostic logging for Dynamic Verifier is on by default and exports logging information to an Intelligent Capture V_ log file. To turn logging off, complete the following steps.

1. Open the ***<project>*.ini** file.
2. Set `GRL_OP_DynamicDebug` to `No`.
3. Save the changes.

About configuring users

You use settings in the BRWUser user table to configure users, including the following settings.

- The Intelligent Capture permissions for each user.
- Whether a user logs in using Windows authentication, or with a username and password.
- The client groups a user can access and the corresponding documents a user can process.
- Which users are subject to quality reviews.

After you configure the user table, you configure a server job to import the user table into Intelligent Capture automatically.

What are the authority level settings?

The AuthorityLevel column controls what the user is permitted to do within the Verifier application. Possible entries are as follows.

AuthorityLevel value	Description
ADM	The Administrator role is to manage users, groups, and user-to-group assignments. Administrators install the system, configure applications, and manage data. They also design and maintain projects. This role is the most powerful of the roles, because it encompasses the permissions for all other authority objects. For a user that is granted the ADM role, the client group may be left blank. If it is left blank, the administrator is able to see documents in Verifier belonging to all clients.
SLM	The Supervised Learning Manager role is to define, modify, and maintain the Learnset. This functionality is accessible only through Verifier.
SLV	The Supervised Learning Verifier role is to collect and manage local training data. Supervised Learning Verifiers are subject-matter experts who can propose Learnset candidates to improve system performance. This functionality is accessible only through Verifier.
VER	The role of the Verifier is to verify documents that could not be automatically processed. Typically, members of the Verifier group are clerks. This functionality is accessible only through Verifier.
SET	The Verifier Settings role is to change the Intelligent Capture Verifier configuration. This role is given to users who are considered to have enough knowledge of the application to make changes that is beneficial to all Intelligent Capture Verifier users.

Populate the user table

The BRWUser table is keyed by a unique combination of users' user names and the client groups to which they are assigned. If a user is assigned to multiple client groups, then multiple rows need to be added into the table. For example, if user JSMITH needs to be assigned to client groups 1 and 2, the UserID column has two entries in the BRWUser table.

To populate the user table, complete the following steps.

1. Open the **BRWUser** table.
2. In the **UserID** column, enter a user's user name.
3. To have the user log in with a username and password, enter a password In the **Password** column. If you want the user to log in using Windows Authentication, leave the column blank.
4. If the user is using Windows Authentication to log in, in the **Domain** column, enter the Windows Domain information.
5. In the **Primary Group Name** column, enter the group name for the group to which the user is assigned.
6. Save the changes.

Configure an automatic import job

After the BRWUser table is populated, an automatic import job is configured to import users into the main database. The automatic import job is configured in the RTS Management Console against the RTS instance that is carrying out document import.

To configure an automatic import job, complete the following steps.

1. Open the **RTS Management Console**.
2. On the **General** tab, complete the following substeps.
    1. In the **Every** field, enter the number of minutes, hours, or days that Intelligent Capture imports users from the **BRWUser** table.
    2. From the time list, select minutes, hours, or days.
    3. In the **Starting at fields**, enter the date and time that Intelligent Capture should start importing users.
    4. To update system security during an import, select **Update system security**.
    5. To update the pool automatically, select **Automatic pool update**.
3. On the **General** tab, click **OK**.

What are the automatic import job errors?

An error is displayed and the import fails if any of the following conditions are met. Error messages are located in the Intelligent Capture log file for the RTS instance performing the user import job.

- You do not supply a connection string for the Intelligent Capture database referenced by the SQL connection group in the GRL section of the <*project*>.ini file.
- The system is unable to connect to the Intelligent Capture database.
- The BRWUser table is empty.
- A user name column is blank.
- The Client Group column is blank and the user is not an administrator.
- The Client Group column does not contain a numeric value.
- The primary Group ID column is not populated.
- You have not allocated clients to the Client group in the BRWClient table.

Appendix A: Configure Institution ID and Applicant ID fields without using a partition

About configuring the Institution ID and Applicant ID fields without using a partition.

This section contains the instructions for configuring the Institution ID or Applicant ID fields in Intelligent Capture without using a partition. The system determines the institution ID or applicant ID with the Intelligent Capture Associative Search engine. Each institution or applicant at a single address must have a unique identifier. This unique identifier can be either numeric or alphanumeric.

What is the CSV file format?

If you are using a flat CSV file, then it must meet the following requirements.

- Each row in the file should represent a single institution or applicant at a single address.
- Each row should include, as a minimum, columns that represent the institution or applicant name, the street address, the city, and the postcode/zip code.
- Each row in the file must have one column that is a unique identifier for that record and is common only to that row.
- Each row in the file must have an equal number of columns.
- The column separator must be a semicolon, such as content within a single column must be stripped of any semicolons in advance.
- Each column must be stripped of double-quotes (“).
- If the CSV file is being used as the institution or applicant extract source, the first line of the CSV file needs to include the column names, and that the Import field names from first row option be enabled for the InstitutionASE or ApplicantASE field in the designer. This configuration eases Verifier searches when using the Details option in the Verifier search.

What is CSV file encoding for non-western characters

If the CSV file includes non-western characters, the file must have a UNICODE encoding. ANSI or UTF-8 encodings are not supported.

About using a Standard Institution or Applicant ID field without a partition

This is the most basic configuration option for configuring an Institution or Applicant ID field without using a partition. You should use this configuration when the SIS system provides a single field identifier for a single institution or applicant at a single address.

In this scenario, each record in the institution or applicant extract is supplied by the client, whether it is provided as a CSV file or within a database table, and it should represent a single institution or applicant at a single address. Additionally, one column in the record needs to be a unique identifier. The generation of the institution or applicant pool based upon the institution or applicant extract fails if more than one record shares the same unique identifier.

Configure Standard Institution ID and Applicant ID fields without a partition

To configure a standard Institution or Applicant ID field, complete the following steps. Once for the InstitutionASE field and once for the ApplicantASE field.

Note If you are using UNC paths, the relevant directories should have the appropriate shares, which is usually full control, so that the system can perform the required read-write operations.

1. Navigate to the directory that contains the **Global project** file.
2. Create a new directory and name it `Pool`.
3. Open the **<*project*>.ini** file, locate the **ASA** section, and complete the following substeps.
    1. Set the Class parameter to **Transcripts**.
    2. Depending on whether you are configuring an **Institution ID** field or an **Applicant ID** field, set the **Fieldname** to **Institution ASE** or **Applicant ASE**.
    3. If the unique identifier for each row in the institution extract is numeric, then set the **AlphaNum** parameter to **No**. Otherwise, set it to **Yes**.
    4. If the **Pool** directory is located in the same directory as the project file, then leave the **PoolPath** parameter empty and set the **PoolRelative** parameter to **Yes**. Otherwise, in the **PoolPath** parameter, enter the path to the pool, and set the **PoolRelative** parameter to **No**.
    5. Set the **PoolDirectory** parameter to **Pool**.
    6. Set the **PoolName** parameter to **Institutions** when configuring an **Institution** field, and set it to **Applicant** when configuring the **Applicant** field.
    7. If the **Institution Extract** file is located in the same directory as the **<*project*>.ini** file, then set the **FileRelative** parameter to **Yes**, and in the **ImportFileName** parameter enter the name of the **Institution Extract** file, including the file extension. If the **Institution Extract** file is not located in the same directory as the **<*project*>.ini** file, then set the **FileRelative** parameter to **No**, and in the **ImportPathFileName** parameter, enter the **UNC** path to the **Institution Extract** file.
    8. Save the changes and close the INI file.
4. Open the **project** file, and in the **Transcripts** class, navigate to the **InstitutionASE** field, display the field settings, and complete the following substeps.

Note The names of the columns shown in the field display are system-assigned names. These names may not be indicative of the contents of the fields in the Institution Extract file. SupplierID is always the first column in the Institution Extract file, SupplierIndex is the second column, SupplierName is the third column, and so on. When using a CSV file as the institution or applicant extract source, we recommend that the first line of the CSV file contain the column names as they appear in the Institution Extract file, and that you enable the Import field names from first row option for the InstitutionASE or ApplicantASE field in the designer. This configuration eases Verifier searches through the Details option within Verifier search.

1. To import the pool, on the **File Import** tab, click **Import**.
2. To configure the search fields to identify the institution or applicant, on the **Analysis** tab, in the **Search** column, select the **Institution name**, **Street address**, **City**, **Zip/postal code**, and **Institution telephone numbers** boxes.
3. To select the column in the institution extract that denotes the unique identifier for the institution record, select the radio button in the **ID** column.

5. To reimport the pool, on the **File Import** tab, click **Import**.
6. Set the **Class** settings to the following configuration.

   ```
   [*institution name*] + underscore + [*institution ID*]
   ```

   However, if the Org_ID column contains the unique institution ID, and the InstName column contains the institution name, which varies depending on the column order in the actual institution extract, then set the class to the following configuration.

   ```
   [InstName]_[Org_ID]
   ```

7. The field settings control how the institution address is displayed on the Verifier form. It is a multi-line field, and the first line must be set to the unique identifier for the record in the institution extract. The file should have the following structure, but the structure may vary depending on your business needs. However, the first line must be set to the unique identifier.

   ```
   [*Unique ID*]
   [*Institution Name*]
   [*Street Address*]
   [*City*], [*State / Region*] [*Postal / Zip Code*]
   ```

8. The configuration of the Institution or Application field is complete. A green light with the message Engine Is Ready should appear in the field status box. Save and close the project file.

Map the field identification

To map the identification of each field in the institution master extract to the relevant fields in the BRWSRC and BRWSRA tables, complete the following step. It is recommended that as many fields as possible are mapped in the institution extract, which must include the ID parameter and all the fields displayed in the Institution Address dialog box.

- In the **BRWSRC** or **BRWSRA** table, set the following fields to the following values.

  ```
  ID=Org_ID

  Name=InstName

  Address 1=Address1

  City=City

  Zip=Postal

  State=State
  ```

About configuring columns for the Institution ID, Applicant ID and Site ID fields

Intelligent Capture does not permit the use of a composite key within the Institution or Applicant file extract. The two values need to be within a single column if you have a single institution, or an applicant at a single address, that is represented in the downstream SIS system by a combination of an institution and applicant ID and a site ID.

The nominated separator is specified within the AlphNumSiteSeparator column in the BRWINS table for an institution and the BRWAPT table for an applicant. This must be populated and adhered to if you are using alphanumeric Institution or Applicant ID fields and a Site ID field. An error is displayed if this configuration is not followed, or more than one separator is found as part of a single unique identifier.

Therefore, in the Institution or Applicant extract file, the following columns are required.

- A column representing the combined unique identifier.
- A column representing the site ID.
- A column representing the institution or applicant ID.

Configure the Institution ID/Applicant ID and Site ID columns

The steps for placing the values in a single column depend on whether the Institution or Applicant ID and Site ID fields are numeric or alphanumeric.

To configure institution/applicant and site ID columns, complete the following steps.

1. If the both fields are numeric, the Institution or Applicant extract file needs to contain an additional column representing the combined institution or applicant ID and site ID, using the following formula.

   ```
   Unique Identifier = (Institution/Applicant ID * 1000000 ) + Site ID
   ```

   For example, if the institution ID is 1234 and the site ID is 5678 then the unique identifier is as follows.

   ```
   (1234 * 1000000) + 5678 = 1234005678
   ```

2. If either the institution or applicant ID or the site ID contains alpha characters, then use the following formula for combining the two.

   ```
   Unique Identifier = Institution/Applicant ID + [Separator] + Site ID
   ```

   For example, if the institution ID is A12345, the Site ID is 1000, and the designated separator is a hyphen (-), then the Institution Extract file should have the following in the unique identifier column.

   ```
   A12345-1000
   ```

Configure the InstitutionASE or ApplicantASE columns

To configure the InstitutionASE and/or ApplicantASE columns, complete the steps in the **Configure a standard Institution or Applicant ID field** without a partition section with the following variations.

1. For **step 8**, set the ID in the **class-name** to the field that represents the standalone institution or applicant ID.
2. For **step 11**, in the **BRWSRC** or **BRWSRA** tables, complete the following mapping.
   - Map the **Unique Record Identifier** field to the **ID** column.
   - Map the **SiteID** field to the **SiteID** column.
   - Map the **Institution ID** or **Applicant ID** field to the **ExternalID** column.

   For example, if the CSV file has allocated technical names of IndexID to the unique ID column, SiteID to the site ID, and Org_ID to the institution ID component, then the mappings are as follows.

Column in BRWSRC	Value
ID	IndexID
SiteID	SiteID
ExternalID	Org_ID

About configuring the External Institution an Applicant ID columns

Use the configuration steps in this section if the downstream-SIS system differentiates between an internal and an external institution or applicant ID by using an internal institution or applicant ID at the database table level, but the user is presented with an external ID through the application itself.

If the client requires that the Verifier application follows this pattern and displays the external institution or applicant ID to the user, then the institution or applicant extract requires that the external institution or applicant ID is included as a column, but the SIS system internal institution or applicant ID is the unique identifier.

Configure the External Institution or Applicant ID columns

To configure an external institution and applicant ID columns, complete the steps in the **Configure a standard Institution or Applicant ID field** without a partition section with the following variations.

1. For **step 8**, set the ID in **class-name** to the field that represents the external institution or applicant ID.
2. For **step 11**, in the **BRWSRC** or **BRWSRA** tables, complete the following mapping.
   - Map the **Unique Record Identifier** field to the **ID** column.
   - Map the **Institution ID** or **Applicant ID** field to the **ExternalID** column.

3. If the downstream uses an external institution ID and a site ID, in the **BRWSRC or BRWSRA** tables, complete the following mapping.

   · Map the Unique Record Identifier field to the ID column.

   · Map the SiteID field to the SiteID column.

   · Map the External Institution field to the ExternalID column.

   For example, if the CSV file has allocated technical names of IndexID to the unique ID column, and Org_ID to the external institution ID component, then the columns in the BRWSRC table are mapped as seen in the following table.

Column in BRWSRC	Value
ID	IndexID
ExternalID	Org_ID

   Or, if the downstream uses an external institution ID and a site ID, then the columns in the BRWSRC table are mapped as seen in the following table.

   Note In this example, Org_ID represents the external institution ID field, rather than the internal institution ID.

Column in BRWSRC	Value
ID	IndexID
SiteID	SiteID
ExternalID	Org_ID

Appendix B: Using review states

What are review states?

PIC for Transcripts has a second verification step that you can turn on or off at either the user or client levels. This step allows for additional quality control of either automatic extractions, or a user entry prior to document export.

Activating this second level of document review changes the document flow to the following.

- If all fields are extracted by the system automatically, then the document flow is as follows.

  Import → OCR → Classification → Extraction → Review → Export

- If one or more fields in the document require a review by a user, then the document flow is as follows.

  Import → OCR → Classification → Extraction → Verification → Review → Export

When a document goes to review, it is set to a specific state, which by default is 699. This state is accessible only by members of your organization who are authorized to review documents. When the reviewers enter the batch through the Verifier, they can make changes to any of the fields if they detect any issues, or if they do not find any issues, they can press Enter on the first editable field. The document then moves to the regular export.

The before and after values for each field are stored in the Visibility reporting database along with the review start and end time, and the ID of the reviewer.

Set the review state

To set the review state, complete the following steps.

1. Open the **<*project*>.ini** file.
2. Navigate to the **GRL** section.
3. Set the **GRL_VL_ReviewState** parameter to `699`.

   Note If you configure any deviating values, set the values within the 650-699 or 701-749 range.

Activate document review for clients

You can activate document reviews at the client level to send every processed document in a specific client to the review state prior to data export. To activate document reviews, complete the following steps.

1. Open the **BRWClient**.
2. Set the **RequiresReview** column to `True`.
3. Save the changes.

About activating document review for specific users

You can activate document reviews at the individual user level. If you implement review states for users, then all documents processed by that user are sent for further review, regardless of the setting at the client level in the BRWClient table. This can be relevant for operations who would wish documents processed by less-experienced users to be subject to supervisory review until the user becomes more proficient.

Activate document review for specific users

To configure document review for a user, complete the following steps.

1. Open the **BRWUser** table.
2. Set the **RequiresReview** column for the user to `True`.
3. Save the changes.

Appendix C: Configure Visibility

About configuring Visibility

Perceptive Visibility is a reporting tool that provides access to solution performance data using a web interface. It is a separate component to Intelligent Capture for Transcripts. The tool contains over 40 standard reports that allow process supervisors to do the following.

- Obtain solution key performance metrics.
- Monitor documents as they move through the system.
- Identify solution bottlenecks.
- Report on productivity at the project and client levels.
- Report on user productivity.

The Perceptive Visibility portal points to a standard table schema held in a reporting database, which is populated by the Intelligent Capture solution as documents move through the system.

This following section describes the steps necessary to activate the connection to the reporting database. As a prerequisite, the reporting database should already have been created, and the scripts required to create the tables executed.

Configure Visibility reporting

To configure Visibility reporting, complete the following steps.

1. Open the **<*project*>.ini** file.
2. Set the **ConnectToReportingDB** parameter to `Yes`.
3. Set the **SQLConnectionGroup** parameter to the following.

   ```
   REP_OP_ConnectToReportingDB=Yes
   REP_VL_SQLConnectionGroup=01
   ```

4. Intelligent Capture begins the reporting trail for each document upon the initial import of the document into the system, but if you want reporting to start sooner, such as at scan time, then set the **REP_OP_StartNewRecordForImportedDocument** parameter to `No`.
5. Set the **ConnectionString** parameter to the following for each string connection, SQL_VL_01_ConnectionString, SQL_VL_02_ConnectionString, and so on.

   ```
   SQL_VL_01_ConnectionString=Provider=SQLOLEDB.1;Password=test;Persist Security
   Info=True;User ID=test;.initial Catalog=PICT;Data Source=W08-SERVER\SQLEXPRESS,1254
   ```

   Note If no SQL connection group is specified, the system always defaults to group 01. This applies to all SQL connection groups within the system.
6. Set the **IMP_VL_URN** parameter to `COMPONENT2`.
7. Set the **REP_VL_ReportingKey** parameter to `URN`.

8. Optional. If you want to use a specific naming convention for any Visibility reporting tables created in the databases, set the names of the tables using the following parameters.

   ```
   REP_VL_ReportingDBDocumentTable=BRWdocument
   REP_VL_ReportingDBFieldTable=BRWdistillerfields
   REP_VL_ReportingDBLineItemsTable=BRWdistillerLineItems
   REP_VL_ReportingDBLineItemsTable2=BRWdistillerLineItems2
   REP_VL_ReportingDBHistoryTable=BRWdocstatus
   REP_VL_ReportingDBImageTable=BRWDOCIMAGE
   ```

9. Optional. Information is not written to a database from any documents processed in the Designer module. However, you can have information written to a database from any document processed in the designer for testing and debugging purposes by setting the **REP_OP_ReportingInDesigner** parameter to `Yes`.

   Note In a production environment, this should always be set to `No`.

10. Optional. If you want to use the Visibility Reporting database to house an image of the document, set the REP_OP_StoreImageInReportingTables parameter to Yes. Only configure this parameter if you are using Perceptive Visibility reporting for late archiving. In all other cases, to display the document image using Visibility, set the REP_VL_StorageDirectory parameter to the following.

    ```
    REP_VL_StorageDirectory=\\My Computer\Perceptive Projects\Export
    ```

11. Optional. To retrieve a document stored within the Perceptive Visibility tables using a URL, which is both stored against the record in the reporting database and is available for export to a downstream system, set the **REP_VL_ArchiveURL** parameter to the following.

    ```
    REP_VL_ArchiveURL=http://archivesystem.perceptive.com/Page.aspx?URN=XXXXX
    ```

    Where `XXXXX` is the point in the URL where the unique document identifier is inserted by the system to retrieve the image.