X9Assist User Guide

X9Ware

Your x9.37+ACH+CPA005 support tools

This document contains user guide information for all of our desktop products:
X9Vision, X9Validator, X9Assist, AchValidator, and AchAssist.

Revision Date: 04/26/2024
Release R5.04

X9Ware LLC
10753 Indian Head Industrial Blvd
St Louis, Missouri 63132-1101
(844) 937-1850

Email support@x9ware.com

Table of Contents

X9Ware License Agreement

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

 This END-USER LICENSE AGREEMENT (EULA) applies equally to all X9Ware LLC products including but not limited to X9Assist, X9Lite, X9Vision, X9Validator, AchAssist, AchValidator, X9Utilities, X9Export, the X9Ware-SDK, the X9Ware-E13B-OCR toolkit, the X9Ware-TiffKit, and all other X9Ware software products that are offered by X9Ware LLC. These are collectively referred to within this EULA as the "SOFTWARE PRODUCT". PLEASE READ ALL TERMS AND CONDITIONS ASSOCIATED WITH THIS LICENSE AGREEMENT CAREFULLY BEFORE CONTINUING WITH THE INSTALLATION OF THIS SOFTWARE PRODUCT. X9Ware LLC End-User License Agreement ("EULA") is a legal agreement between you (either as an individual or as a business entity) and X9Ware LLC (referred to as "licensor"), for your use of the above identified product(s) and associated materials. By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. This EULA represents the entire agreement concerning the use of the SOFTWARE PRODUCT between you and X9Ware LLC, and it supersedes any prior proposal, representation, or understanding between the parties. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT. If future changes made to this EULA are due to new features or functionality in the Software, your use of such features demonstrates acceptance of changes to the EULA with respect to such features and functionality. Licensee may use the Licensed Programs in executable format for their internal use. The SDK may be used as the basis for the development of other software. Licensee may not, however, transfer or sub-license the Licensed Programs to any third party, in whole or in part, in any form, whether modified or unmodified.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties and is made available subject to the specific terms of this EULA. X9Lite is our freeware x9 viewer. All other X9Ware products are NOT FREE SOFTWARE and must be explicitly licensed by each and every user.

THIS AGREEMENT is automatically applicable when accepted as part of our software installation click-through process and is binding based on that acceptance.

THIS AGREEMENT may also be signed and submitted as a hard copy document. When initiated in that matter, it is entered into as of _____("the date of the last signature below ("Effective Date") by and between X9Ware LLC, with offices at 10753 Indian Head Industrial Blvd, St Louis, Missouri 63132-1101, ("X9Ware" or "LICENSOR") and _____ ("LICENSEE" or "You" or "Your") at address _____. X9Ware and You are referred to in this Agreement individually as a "party" or collectively as the "parties."

1 .GRANT OF LICENSE

a) Installation and Use. Subject to the terms of this EULA, you are hereby licensed by X9Ware to use the SOFTWARE PRODUCT on your workstations, computers, or electronic devices for the defined duration of your license agreement.

b) License Duration and Licensing Key. Licenses are typically issued for the term of one (1) year (twelve months), but may be issued for varying terms depending on unique customer requirements, at the request of the user. In support of this EULA, you will be issued a license key which is unique to a company or user name, the licensing level, the license expiration date, and the program release. The license key must be used only by the party it has been issued to and must not be shared with others. The Licensee must protect the confidentiality of the license key with the same care as with any of their own proprietary data. Your license key is activated after making confirmed payment to X9Ware. You may distribute the license key within your organization as needed in accordance with your licensing level. This licensing agreement does not cover private label (branded) solutions, where the intent is distribution of the SOFTWARE PRODUCT to your clients and customers. Any such licensing arrangement will require more extensive discussions and agreements between both parties.

2. CONSIDERATION TO X9Ware

a) License Fees. Licensee shall pay, upon delivery of the Licensed Programs, the license fees set forth per Exhibit A which may be optionally attached.

b) Other Fees. License fees do not include any duties, bank fees, sales, use, excise or similar taxes due. If Licensor is required to pay any such amounts, Licensee shall reimburse Licensor in full.

3. OWNERSHIP

The original and any copies of the Licensed Programs, made by Licensee, including translations, compilations, partial copies, modifications, and updates are the property of X9Ware.

4. PROPRIETARY RIGHTS

a) Proprietary Information. Licensee recognizes that X9Ware regards the Licensed Programs as its proprietary information and as confidential trade secrets of great value.

b) Confidentiality.

(i) Definitions. By virtue of this Agreement, a party ("Receiving Party") may have access to information from the other party or its affiliates ("Disclosing Party") that would reasonably be considered proprietary or confidential, including but not limited to the Disclosing Party's or its affiliates' employees, customer lists and data, financial information, technical information, business plans, organization, policies, or products ("Confidential Information"). Confidential Information shall include any written reports, findings, conclusions, recommendations, or reporting data and analysis prepared by Disclosing Party and provided to Receiving Party under this Agreement, or information that due to its nature or the circumstances of its disclosure, a person exercising reasonable business judgment would understand to be confidential or proprietary of a party.

(ii) Exceptions to Nondisclosure. This nondisclosure provision shall not apply to that information which: (i) is or becomes generally available to the public other than as a result of disclosure by the receiving party; (ii) the disclosing party agrees in writing to make available to the receiving party on a non-confidential basis; (iii) is received from a third party free to disclose such information without restriction; (iv) is independently developed by the receiving party without the use of any of the Information received from the disclosing party; (v) is required by law or regulation to be disclosed, but only to the extent and for the purposes of such required disclosure; (vi) is known to the other party prior to its disclosure by either party; or (vii) is disclosed in response to a valid order of a court or other governmental body of the United States or any political subdivisions thereof, but only to the extent of and for the purposes of such order, and only if the receiving party first notifies the disclosing party of the order and permits the disclosing party to seek an appropriate protective order.

(iii) Use and Nondisclosure. Receiving Party agrees to hold Confidential Information in confidence. Receiving Party agrees not to collect or use Confidential Information for any purpose other than for the implementation of and business purposes expressly set forth in this Agreement. Receiving Party further agrees not to make any Confidential Information available in any form to any third party, except that such information may be disclosed to the parties' affiliates, attorneys, accountants, agents, employees, contractors and consultants (collectively the "Representatives") on a need-to-know basis only and under obligations of confidentiality at least as restrictive as those contained herein.

(iv) Degree of Care. Receiving Party agrees to use the same degree of care that it uses to protect its own confidential information of a similar nature and value, but in no event less than a reasonable standard of care, to ensure that Confidential Information are not disclosed or distributed by its Representatives in violation of the provisions of this Agreement. Such care shall include, but not be limited to, Receiving Party's maintenance of appropriate administrative, technical and procedural safeguards to: (i) ensure the security and confidentiality of Confidential Information, (ii) protect against any anticipated threats or hazards to the security or integrity of Confidential Information, and (iii) protect against unauthorized access to or use of Confidential Information which could result in substantial harm or inconvenience to Disclosing Party. Receiving Party represents that it has required agreements with its representatives (those who may have access to any Confidential Information) which enable it to comply with the terms of this section. Notwithstanding the foregoing, Receiving Party shall be responsible for any breaches of this nondisclosure provision by any of its Representatives. Receiving Party shall promptly report to Disclosing Party any unauthorized disclosure or use of Confidential Information by Receiving Party or its agents of which Receiving Party becomes aware.

(v) Additional Obligations. Unless agreed otherwise in writing, any personally identifiable information provided by Disclosing Party to Receiving Party or learned by Receiving Party hereunder may only be used for conducting the business transactions that are the subject of the Agreement. Disclosing Party does not consent to Receiving Party's use of any personally identifiable information provided by Disclosing Party or any of its Representatives, for any other purpose than for conducting business transactions that are subject of the Agreement.

(vi) Length of Obligation and Return of Information. Receiving Party shall at any time upon the Disclosing Party's written request either return or destroy, at the Disclosing Party's option, the Disclosing Party's Confidential Information. Notwithstanding the foregoing, Receiving

Party may keep a copy of Confidential Information solely for maintaining reasonably appropriate business records or as may be required by law. Confidential Information shall be subject to these provisions for five (5) years after termination of the Agreement.

(vii) Unauthorized Access or Use. Receiving Party shall: 1) promptly notify Disclosing Party of any material unauthorized possession, use or knowledge, or attempt thereof, of Confidential Information by any person or entity which may become known to it; 2) promptly furnish to Disclosing Party full details of the unauthorized possession, use or knowledge, or attempt thereof, and assist Disclosing Party in investigating or preventing the recurrence of any unauthorized possession, use or knowledge, or attempt thereof, of Confidential Information; 3) cooperate with Disclosing Party in any litigation and investigation against third parties deemed necessary by the other party to protect its proprietary rights; 4) promptly use its best efforts to prevent a recurrence of any such unauthorized possession, use or knowledge, or attempt thereof, of Confidential Information; and 5) bear the cost it incurs as a result of compliance with this section.

5. COPIES

a) Backup Copies. You may make copies of the SOFTWARE PRODUCT as may be necessary for backup, archival, and disaster recovery purposes.

b) Distribution. You may make copies of the SOFTWARE PRODUCT as may be necessary for internal distribution within your business and solely for that purpose. The SOFTWARE PRODUCT is copyrighted but unpublished by X9Ware. Licensee agrees to reproduce and apply the copyright notice and proprietary notice of X9Ware to all copies made hereunder, in whole or in part and in any form, of Licensed Programs.

c) Seat Tracking. Certain X9Ware products are licensed at the seat or device level. For these specific products, Licensee recognizes that their internal usage must be restrained to these limits per the product which has been purchased. Licensee agrees to take reasonable efforts to maintain records of the number of installed seats or devices and upgrade to an expanded license should their usage exceed their licensed seating level.

6. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

a) Maintenance of Copyright Notices. You must not remove or alter any copyright notices on any and all copies of the SOFTWARE PRODUCT.

b) Distribution. You may not distribute registered copies of the SOFTWARE PRODUCT to third parties. Evaluation and production versions of the SOFTWARE PRODUCT are available for download directly from X9Ware.

c) Prohibition on Reverse Engineering, Decompilation, and Disassembly. You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation. Java classes, methods, data structures, and algorithms are the property of X9Ware. Use, sharing, or distribution of The Software in any other manner is prohibited without prior written permission from X9Ware.

d) Rental. You may not rent, lease, re-assign, or lend the SOFTWARE PRODUCT.

e) Support Services. X9Ware may provide you with support services related to the SOFTWARE PRODUCT ("Support Services"). Any supplemental software code provided to

you as part of the Support Services shall be considered part of the SOFTWARE PRODUCT and are subject to the terms and conditions of this EULA.

f) Compliance with Applicable Laws. Each party shall comply with all federal, state and local laws and regulations applicable to that Party with respect to the business transactions that are the subject of this Agreement.

7. NOTICES

All notices in connection with this EULA shall be in writing and may be given by certified, registered, or first class mail or personally delivered at the receiving party's address set forth on the front page. For purposes of this EULA, a notice shall be deemed effective upon personal delivery to the party or if by mail five (5) days after proper deposit in a mail box.

8. SUCCESSORS

This EULA will be binding upon and will inure to the benefit of the parties hereto and their respective representatives and successors except as otherwise provided herein.

9. SEVERABILITY

In the event any provision of this EULA is determined to be invalid or unenforceable, the remainder of this EULA shall remain in force as if such provision were not a part.

10. TERMINATION

a) Termination. X9Ware may terminate this EULA if the customer defaults on any of the terms and conditions as stated within this EULA, and then fail to correct such conditions within thirty (30) days after written notice thereof from X9Ware. Additionally, this contract with all terms, provisions, liabilities, and warranties are automatically cancelled when a license or maintenance agreement is not renewed, or that renewal payment is more than 60 days past due. If this occurs, provisions of this contract will be automatically reinstated if X9Ware accepts a subsequent late payment which brings payment status into a current state. Acceptance of such a late payment is at the discretion of X9Ware.

b) Discontinuance. In the event of termination, Licensee will immediately discontinue use of the Licensed Programs. Within one (1) month after termination of this EULA, Licensee will furnish to X9Ware written correspondence which certifies with respect to each of the Licensed Programs that, through its best effort and to the best of its knowledge, the original and all copies, in whole or in part and in any form, of each of the Licensed Programs have been destroyed.

11. INTELLECTUAL PROPERTY

a) Ownership. All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by X9Ware.

b) Intellectual Property. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by X9Ware.

12. SCOPE

a) Scope. This Agreement shall be governed by the laws of the State of Delaware without regard to conflict of laws principles. In the event that a dispute arises between the Parties which results in legal action, the successful Party shall be entitled to be reimbursed for its costs and expenses therefor, including reasonable attorneys' fees.

b) Prior Agreements. This License contains the entire agreement between the parties with respect to the subject matter hereof, and supersedes all prior agreements or understandings (either oral or written).

13. WARRANTY DISCLAIMER

a) As Is. X9Ware licenses, and Licensee accepts, the licensed programs "AS IS." X9Ware PROVIDES NO WARRANTIES AS TO THE FUNCTION OR USE OF THE LICENSED PROGRAMS, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LICENSED PROGRAM IS WITH LICENSEE. LICENSOR DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE LICENSED PROGRAMS WILL MEET LICENSEE'S REQUIREMENTS OR THAT THE OPERATION OF THE LICENSED PROGRAMS WILL BE UNINTERRUPTED OR ERROR FREE.

b) No Warranties. X9Ware does not warrant or assume responsibility for the accuracy or completeness of any functionality, accuracy, information, text, graphics, or other items contained within the SOFTWARE PRODUCT. X9Ware makes no warranties respecting any harm that may be caused by the transmission of computer viruses or other such malicious computer programs. X9Ware further expressly disclaims any warranty or representation to Authorized Users or to any third party.

14. LIMITATION OF LIABILITY

a) Damages. In no event shall either party be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) arising out of 'Authorized Users' use of or inability to use the SOFTWARE PRODUCT, even if X9Ware has been advised of the possibility of such damages.

b) Liability. In no event will either party be liable for loss of data or for indirect, special, incidental, consequential (including lost profit), or other damages based in contract, tort or otherwise. In no event shall either party shall have liability with respect to the content of the SOFTWARE PRODUCT or any part thereof, including but not limited to errors or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, loss of privacy, or the disclosure of confidential information.

15. EXPORT REGULATIONS

a) Regulatory Agencies. Licensee understands that X9Ware is subject to regulation by agencies of the U.S. Government, including the U.S. Departments of Commerce and State, which prohibit export or diversion of certain technical products to certain countries.

b) Export. Licensee warrants that it will comply in all respect with the export and re-export restrictions set forth in the export license for the Licensed Programs and all other applicable export regulations.

c) Indemnification. Each party agrees to indemnify and hold the other harmless from any loss, damages, liability or expenses incurred by a party as a result of the other party's failure to comply with any export regulations or restrictions.

16. ENTIRE AGREEMENT

This EULA sets forth the entire understanding between the parties with respect to the subject matter hereof, and merges and supersedes all prior agreements, discussions and understandings, express or implied, concerning such matters. This EULA shall take precedence over any additional or conflicting terms which may be contained in Licensee's purchase order or X9Ware's quotes, invoices, or order acknowledgment forms.

17. Acceptance

By:	**By: X9Ware**
Name:	Name:
Title:	Title:
Signed:	Signed:
Dated:	Dated:

Third Party Licenses and Copyrights

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

X9Ware LLC Copyright and License Agreement

Use of Third Party Products

Certain third party products are used by X9Ware for specific functions. Each of these products have their own respective licenses as follows:

Product	X9Ware Usage	Applicable License
Zulu OpenJDK	OpenJDK is a free and open-source implementation of the Java SE Platform Edition. OpenJDK is developed by Oracle, OpenJDK, and the Java Community. However, the top-notch companies like Red Hat, Azul Systems, IBM, Apple Inc., SAP AG also take an active part in its development.	GNU GPL version 2.
Oracle (Sun Microsystems) JAI	A subset of the Sun JAI is used for various TIFF image capabilities.	The Copyright statement is provided below. JAI is protected by the Apache 2.0 License: http://www.apache.org/licenses/LICENSE-2.0
Oracle (Sun Microsystems) SwingX	A subset of the Sun SwingX UI extensions are used by X9Assist, as is, with no modifications.	The Copyright statement is provided below. SwingX is protected by the Apache LGPL license.

		https://www.gnu.org/licenses/lgpl-3.0.en.html
Apache POI	Apache POI is used for Excel spreadsheet read and write support.	Apache POI is protected by the Apache 2.0 License. http://www.apache.org/licenses/LICENSE-2.0
Apache Commons	Several Apache Commons tools are used including Apache Commons Lang3, Apache Commons IO, and Apache Commons Collections.	Apache Commons is protected by the Apache 2.0 License. http://www.apache.org/licenses/LICENSE-2.0
SLF4J	The SLF4J logging facade is an abstraction for several logging frameworks.	SLF4J source code and binaries are distributed under the MIT license. https://www.slf4j.org/license.html
JSON-Simple	JSON-Simple is used internally to parse JSON result strings.	JSON-Simple is protected by the Apache 2.0 License. http://www.apache.org/licenses/LICENSE-2.0
MigLayout	MigLayout is used internally for GUI component layouts.	MigLayout is protected by the BSD license which is provided below.
FlatLaf	FlatLaf is used for our cross-platform lite and dark themes.	FlatLaf is protected by the Apache 2.0 License. http://www.apache.org/licenses/LICENSE-2.0
JDatePicker	JDatePicker is used internally for date selection drop downs.	JDatePicker s protected by the BSD license which is provided below.

Open Source Fonts	Certain open source fonts are used for image drawing. All such fonts are freely available for commercial use.	Several of these fonts are protected by the Open Font License. https://opensource.org/licenses/OFL-1.1
Advanced Installer is used to build our Windows installers for X9Assist and X9Utilities	Advanced Installer is used to package and deploy our several of our products for the Windows environment. These include our desktop products (eg, X9Assist) and X9Utilities.	Advanced Installer is a very commonly used tool throughout the industry, for this specific purpose.

Sun Copyright

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that Software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

MigLayout License

License (BSD):

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the MiG InfoCom AB nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Keyboard Shortcuts

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

Icon	Usage	Shortcut Key
	Position on previous bundle.	Control+F6
	Position on the previous x9 record or record group.	Control+F7
	Position on the next x9 record or record group.	Control+F8
	Position on next bundle.	Control+F9
	Position as closely as possible on a specific x9 record based on the x9 record number within the overall x9 file.	

Keyboard Shortcuts

Other shortcut keys are available from the main dashboard as follows:

- F1: help
- Control+Delete: can be used during a modify session to easily delete records or bundles

Menu Bar Shortcuts

Many program functions have associated short cut keys using some combination of Control, Alt, or Shift. Please reference the drop down menus for the specific key sequences that are available.

Button Shortcuts

Virtually all buttons have associated short cut keys, with that specific letter underlined within the button text. For example, Control+X can typically be used for E**x**it.

Quick Start Guide

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

This document is intended as a "quick start" guide for new users of our desktop products (X9Validator, X9Assist, AchValidator, and AchAssist). It most certainly does not cover all things that can be done by these desktop applications. It is instead meant to provide a list of steps needed to perform some of the most commonly used functions. This guide does provide some good ideas regarding some of the more complex functions that can be performed by our more advanced desktop applications. X9Validator / AchValidator users should consider an upgrade to X9Assist/AchAssist should you need thee more advanced tools that are available in those products.

In addition to this quick start guide, most functions include a help button in the upper right corner of their panels, which can be used to launch our user documentation on the currently active task. This documentation can be referenced for more detailed information. We do hope that this document proves to be helpful, and please contact us for suggested improvements.

To launch detailed help information:

1) Within any active tool, use the help button (question mark icon) in the upper right hand corner of the associated panel.
2) For overall help, select / help / help / from the toolbar, or press F1. This will present a list of all available help topics.
3) Browse the list of available help topics and double click to launch.

To launch details about a specific item in an existing file:

1) Open the file.
2) You will see the file opened in the tree viewer in the left-side dashboard panel.
3) +/- boxes can be used to expand individual items within the viewer tree.
4) Clicking an item once will launch it in the field viewer.
5) Clicking an item twice will launch it in the item viewer.
6) Alternatively, the item viewer button can be used to immediately open the item viewer for the currently displayed entry.
7) Alternatively, the thumbnail images (in the lower right) can be used to open the item viewer for the currently displayed entry.
8) The item viewer provides easy access to all information for the current item. It also includes next/previous buttons to let you easily move throughout the file.
9) You can exit from the item viewer at any time using the cancel button, which appears at the bottom of that panel. When you exit the item viewer, you are automatically positioned on the current item within the left-side viewer tree.

10) More information is provided via the “navigation” tab which is available on the viewer tree which appears on the left side of the main panel.

To position on a specific item using the record number:

1) Open the file.
2) On the toolbar, enter the desired record number, where records are numbered sequentially beginning at the file header and ending at the file trailer.
3) Alternatively, on the toolbar, enter the item number, where items (debits and credits) are numbered sequentially within this file.
4) After entering the record or item number, hit the enter key to reposition the viewer.

To find a specific item:

1) Open the file.
2) Issue a find using either / find / find / from the toolbar, or using the binoculars.
3) Enter the information that you have available that describes the item.
4) Hit the find button at the bottom of the panel.
5) The item will now be located based on these attributes.
6) You will get a popup error message if no such item is found.
7) When items are found, the horizontal binoculars image on the tool bar will be formatted as “x/y”, where “x” will initially be one and “y” will tell you how many such items meet your criteria.
8) For example, this might say “1/5”, which means that the viewer is now positioned on the first item and there are a total of 5 that meet your criteria.
9) You can now click the horizontal binocular icon (as a button) to move through the items.

To remove a record from an existing file:

1) Open the file.
2) Turn on “modify” using the check box in the upper right hand corner of the main panel.
3) Locate the item you want to delete in the tree on the left. You can either scroll to it or use find (binoculars) on the tool bar.
4) With the item highlighted, you then do / tools / delete /.
5) You get a popup panel for the delete. It is populated for delete by range and for the item that was selected in the viewer.
6) Review the deletion.
7) Press the delete button.
8) You can a confirmation popup.
9) Select delete and update trailers.
10) Review your resulting file.
11) Save using / file / saveas /.
12) Assign a new name (do not overwrite the original file).

13) You get the option to save an audit trail file – you can either save the log or cancel.
14) The file will be saved and the then reloaded in the viewer.
15) You can now review the totals of the new file and browse all content.

To change a field within an existing file:

1) Open the file.
2) Turn on "modify" using the check box in the upper right hand corner of the main panel.
3) Issue a search either / find / search / from the toolbar, or using the magnifying glass.
4) Enter the search string with the specific value you are looking for along with the "Match exactly" option. If you instead want to selected all values for this field, select the "Replace All" option will will select all values for the selected field type for replacement.
5) On the right-side panel, click the "Select specific" box which will unselect all fields.
6) Scroll through the list of fields and select the desired record and field.
7) On the left-side panel, now choose the replacement strategy. Typically, this will be "Explicit value" to allow a user entered value to be inserted into the chosen field, or will be "blanks" to assign a blank value (which will essentially clear any value that might be currently present).
8) If "Explicit value" has been chosen, then enter the replacement string to be assigned.
9) Hit the search button at the bottom of this panel.
10) Now review the result panel which is shown on the right, which will indicate how many matches have been found based on the entered criteria.
11) If this is correct, you will now typically use the "Replace All" button at the bottom to initiate the replacement of these values across all targeted records.
12) You can alternatively do a selective "replace" action (on a field by field basis), when you do not want to apply a blanket replacement across all targeted records.
13) Exit using the button at the bottom of this panel.
14) A new file validation will be run with the field changes that have been made.
15) The / tools / modification log / function can be used to now review all changes that have been applied, and to actually individually revert any of the changes which have been made.
16) Review your resulting file.
17) Save using / file / saveas /.
18) Assign a new name (do not overwrite the original file).
19) You get the option to save an audit trail file – you can either save the log or cancel.
20) The file will be saved and the then reloaded in the viewer.
21) You can now review the totals of the new file and browse all content.

To export all items from an existing file to Excel:

1) Open the file.
2) Issue / print / excel exporter / from the toolbar to initiate the excel exporter.
3) Click the "All tables" box as needed to get all tables into an unselected mode.
4) Click the "Item table" box to select items.
5) Make sure that "all items" is then selected which will select all items in the file.

6) Hit the create button at the bottom of the panel.
7) Our excel exporter will now run and create an intermediate file with your results.
8) The intermediate file will be launched.
9) Review these results as needed.
10) Optionally save to your permanent output file.

To print MICR checks for an entire file (x9 only):

1) Open the file.
2) Use the / print / print images / function to initiate the print.
3) Select print from file (which is the default).
4) Using the right-side panel, the "load a saved print format" button should be used to load the "Letter 3 x 1 MICR Print.xml" definition which contains parameters to MICR checks using a layout that has three checks per page (which is the most common printer layout).
5) The selected layout will be loaded and will automatically assign print options (on the left) and layout (on the right).
6) Use the preview button (at the bottom) to take a quick look at how items will be printed.
7) If you want, you can optionally now use the view button (at the bottom) to reformat these items into XPS printer format and load the Microsoft viewer. This process is a bit time consuming. It is available in Windows only. This viewer has advanced functionality and also allows pages to be directly printed.
8) Complete your verification of the items to be printed.
9) Use the "print" button at the bottom of this panel to initiate printing.
10) Select your printer.
11) Output will be formatted and routed to your printer.
12) Click OK when completed.

To print IRDs for an entire ICLR (Returns) File (x9 only):

1) Open the ICLR file.
2) Double check your item count.
3) Review the content of the opened file to ensure that the items are as expected.
4) From the tool bar, select / Print / Print IRDs/ .
5) The Print ICL dialog will be opened.
6) Enter your Creator routing number.
7) Enter your Creator business date (it will be defaulted to the current date).
8) Use the "Format" button at the bottom of the panel to initiate formatting.
9) Confirm the popup question to proceed.
10) A Print Image panel will be displayed.
11) Make sure the source is "From Filter" and the filter name is "PrintList".
12) Using the right side panel, load print format "Letter 3 x 1 IRD Print".
13) When the print format is loaded, the print options (attributes) and layout will be automatically updated based on this format definition. You can review this information but it does not need to be entered.

14) Optionally use the Preview button (at the bottom) to take a look at the print format of the items that will be printed.
15) Once the preview is completed, use cancel to return to print.
16) When ready, use the Print button (at the bottom) to initiate the print process.
17) Select your printer.
18) Print.

To print specific Return IRDs within an ICL (Forward) File (x9 only):

1) Open the ICL file.
2) Double check your item count.
3) Review the content of the opened file to ensure that the items are as expected.
4) The following sequence of steps will apply to each IRD to be printed:
    - Locate the next item to be returned within the file tree; if necessary, use the FIND command on the toolbar, which can be accessed using the binoculars icon.
    - Double check you have the right item. If you are adding a single item, then make sure you have selected that one specific item in the file tree. If you want to return an entire bundle, then make sure you have selected that batch headder.
    - With the item to be returned selected within the file tree on the left, from the toolbar use / Filters / Add to Item ReturnsList / to bring up the Returns dialog box.
    - Enter the creator routing number, business date, and item sequence number.
    - Use the drop-down box to select the appropriate return reason.
    - Hit the apply button to add this item to the ReturnsList.
5) Once finished, click the “ReturnsList” button with the far right functional list on the dashboard. This will display all items that have been selected to be printed. Items can be removed from this list as necessary, and additional items can be added.
6) Once the review of the returns list is complete, you are ready to print the IRDs.
7) From the tool bar, select / Print / Print Images / .
8) The Print ICL dialog will be opened.
9) Make sure the source is “From Filter” and the filter name is “ReturnsList”.
10) Using the right side panel, load print format “Letter 3 x 1 IRD Print”.
11) When the print format is loaded, the print options (attributes) and layout will be automatically updated based on this format definition. You can review this information but it does not need to be entered.
12) Optionally use the Preview button (at the bottom) to take a look at the print format of the items that will be printed.
13) Once the preview is completed, use cancel to return to print.
14) When ready, use the Print button (at the bottom) to initiate the print process.
15) Select your printer.
16) Print.

To create a new file using Make:

1) Launch make from the toolbar (/ tools / make).
2) Select the type of file to be created (x9.37 or ach).

3) Press the “load reformatter” button on the left to load our default reformatter. The reformatter is ultimately something that you create, but we provide a sample that you can use as the basis that you can build on. A reformatter definition will transform a use case file (which is either in Excel or CSV format) into logical items. An example (for x9.37) is to load reformatter “depositReformatter.xml” which is a good basis for a reformatter and can be further modified based on your requirements.
4) Press the “load generator” button on the left to load our default generator. The generator is ultimately something that you create, but we provide a sample that you can use as the basis that you can build on. A generator definition will transform the items that are created by a reformatter into an actual output file. An example (for x9.37) is to load “default937.generator.xml” which provides insight into the definitions that are available to you. As you hover over the Make tabs on the right, you will see a popup tool tip which informs you as to whether the information in this tab is saved either as part of the reformatter or as part of the generator. Other than the reformatter tab itself (which maps the columns needed for use case transformation), virtually everything else will be saved within the generator.
5) Press the “load use case” button on the left to load our default use case file. The use case file is ultimately something that you create, but we provide samples that you can use as the basis that you can build on. As an example (for x9.37) you can load “depositFile.xlsx” which contains data that maps into the same columns that are defined by our “depositReformatter.xml” example which is noted above. Remember that you can customize the reformatter definition with any rows that are needed per your requirements. Finally, note that each row within the use case file represents a single item. You can turn any given row into a comment (and not a data row) by setting the COMMENT value for that row to an asterisk. This is an easy way to enable to disable any given row. Even though our use case example file is XLSX, these files can also be XLS and even CSV. The use case file simply represents a series of items, where each item is a single row and each column is defined as a specific data content per the REFORMATTER tab assignments.
6) The title line (at the top) will show how many items have been loaded.
7) Pay special attention to the Errors tab on the right. It will contain information regarding any rows (items) that contain data which has been rejected during our initial inspection and validation process. These items will be excluded from the Make process. If you want to include these items anyway, perhaps as negative use cases which will purposefully force invalid data scenarios for testing, then you must check the “Accept invalid use case data” at the top of the panel; this must be done BEFORE the use case file is loaded.
8) Finally, press the “make” button at the bottom to create your new file.
9) Reformatter and generator definitions can be saved to your own user defined names for subsequent reuse in future X9Assist sessions.
10) Reformatter and generator settings are sticky within the same X9Assist session, even when they are not saved. However, they will be lost for your next X9Assist session if not saved.

To search/replace a specific field value within an existing file:

1) Open the file.
2) Turn on “modify” using the check box in the upper right hand corner of the main panel.

3) Locate the item you want to delete in the tree on the left. You can either scroll to it or use find (binoculars) on the tool bar.
4) With modify enabled, you will see a column within the viewer that is called “modify” and it has a button. This button is used to modify that particular record.
5) Press the modify button for the specific record you want to change.
6) You will get a modify popup that contains the record to be modified.
7) There are current value and new value columns.
8) Change values as needed in the “new value” column.
9) Hit the modify button at the bottom of that panel to modify the current record.
10) You will get a popup asking if you want to update trailer records – respond yes.
11) The field that you modified will now be highlighted in the viewer.
12) Review your resulting file.
13) Save using / file / saveas /.
14) Assign a new name (do not overwrite the original file).
15) You get the option to save an audit trail file – you can either save the log or cancel.
16) The file will be saved and the then reloaded in the viewer.
17) You can now review the totals of the new file and browse all content.

To create an item filter based on criteria:

1) Open the file.
2) Issue a find using either / find / find / from the toolbar, or using the binoculars.
3) Enter the information that you have available that describes the item.
4) Assign an optional filter name (you can skip this and just accept the assigned default).
5) Hit the filter button at the bottom of the panel.
6) Review the resulting filter results, which are shown in a viewer panel.
7) Individual items can be viewed or removed from the filter using item level buttons.
8) The filter can then be used for other purposes such as print image, create new file, or delete items.
9) Close (delete) the filter when no longer needed.

To create a new file with certain records from an existing file:

1) Open the file.
2) Locate each item that is to be included and then use the / filter / add to item PickList / function from the toolbar to add this item to the PickList; you can also do this using the start button on the toolbar.
3) Continue to add items as needed to the PickList.
4) Use the filter viewer to ensure that all selected items are correct; the view and remove buttons can be used to update the filter as needed.
5) Review the selected items in the filter viewer.
6) Use the / filter / create file from items / function to initiate the creation of a new file.
7) Select the “create a new file directly from the selected items” option with the input being the current PickList.

8) Use the create button at the bottom to initiate the file creation process.
9) Review your resulting file.
10) Save using / file / saveas /.
11) Assign a new name (do not overwrite the original file).
12) You get the option to save an audit trail file – you can either save the log or cancel.
13) The file will be saved and the then reloaded in the viewer.
14) You can now review the totals of the new file and browse all content.

To create reversals for an entire file:

1) Open the original file.
2) Review this file and totals to ensure that this is the file to be reversed.
3) Use the / filter / create reversal file from items / function to initiate the creation of the new file.
4) The reversals panel will be displayed and you will see that the input source is automatically assigned as “current file”, since no picklist currently exists.
5) Use the create button at the bottom to initiate the reversal file creation process.
6) Review your resulting file.
7) Save using / file / saveas /.
8) Assign a new name (do not overwrite the original file).
9) You get the option to save an audit trail file – you can either save the log or cancel.
10) The file will be saved and the then reloaded in the viewer.
11) You can now review the totals of the reversal file and browse all content.
12) Repeat this confirmation with a second associate.

To create reversals for individual items:

1) Open the original file.
2) Review this file and totals to ensure that this is the file to be reversed.
3) Find and position on the first item to be reversed. Positioning means that the item is selected within the tree on the left side of the X9Assist panel.
4) Hit the “Star Symbol” on the toolbar, which is used to add the currently selected item to the picklist. You will notice that the picklist is initially created and this item has been added.
5) Repeat these steps to add all of the items to be reversed.
6) Review the items within the picklist to ensue that these are the ones to be reversed.
7) Use the / filter / create reversal file from items / function to initiate the creation of the new file.
8) The reversals panel will be displayed and you will see that the input source is automatically assigned as the picklist, since it’s presence was identified.
9) Use the create button at the bottom to initiate the reversal file creation process.
10) Review your resulting file.
11) Save using / file / saveas /.
12) Assign a new name (do not overwrite the original file).
13) You get the option to save an audit trail file – you can either save the log or cancel.

14) The file will be saved and the then reloaded in the viewer.
15) You can now review the totals of the reversal file and browse all content.
16) Repeat this confirmation with a second associate.

Toolbar

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

The following functions are available from the tool and menu bars:

Tool	Usage
Open	Open a new x9 file, load into memory, and perform all x9 and tiff validations. Note that the initial (default) folder for open operations can be defined in program options.
Close	Close the current x9 file and reset all panels.
Save	Save the current x9 file to the same file name as was originally opened. The file that is saved will include any modifications that you may have made to individual fields or records within the file. As part of the save, you will be prompted to create an optional backup of the original x9 file.
Save As	Save the current x9 file to an external disk file with a new name of your choosing . The file that is created will include any modifications that you may have made to individual fields or records within the file. Save As will allow you to specific the attributes of the output x9 file, which can be different than the input x9 file. For example, the input x9 file could be Ascii without field zero lengths, and the created x9 file could alternatively be Ebcdic with field zero lengths.
Find	Find a specific item based on your selection criteria. Find allows you to enter new selection criteria to begin a find operation. Your previous find criteria is initially displayed so you can easily adjust that criteria to begin your next search. You can create a filter to allow you to easily browse your selection results and then use the selected records for such further activities as creating a new x9 file with just those items or removing the filtered items from the current x9 file.
Find Next	Find the next check based on your selection criteria. Notice that this icon will include information about how many items meet your selection criteria and you relative position within the selection list.

Tool	Usage
Search	Search for specific values and conditions across all record types and then selectively for specific fields within specific record types. You can create a filter to allow you to easily browse your selection results and then use the selected records for such further activities as creating a new x9 file with just those items or removing the filtered items from the current x9 file.
Export	Export one or more of the X9Assist tables tables into Excel or a compatible spreadsheet program such as Open Office. You will first select which of the available tables should be exported. An XLS workbook will then be created that contains the selected tables, where each table will be a separate sheet within the workbook. The XLS spreadsheet is created in the /temp/reports folder. Each new workbook is created with a uniquely defined name through a date and time suffix that is appended to the file name. You can perform various functions with the generated XLS file once it is created. For example, you can print it, share it with others, or save it for future reference. The retention period (in days) for reports is a program option and can be modified based on your requirements. Spreadsheets created in the reports folder will be automatically deleted when this number of retention days is exceeded.
Print Tables	Print one or more of the X9Assist tables. You will first select which of the available tables should be formatted for printing. A PDF file will then be created that contains the selected tables. The PDF file is created in the /temp/reports folder. Each new print file is created with a uniquely defined name through a date and time suffix that is appended to the file name. You can perform various functions with the generated PDF once it has been created. For example, you can print it, share it with others, or save it for future reference. The retention period (in days) for reports is a program option and can be modified based on your requirements. PDF files created in the reports folder will be automatically deleted when this number of retention days is exceeded.
Print Images	Print one or more images from the current x9 file. You can select the current check being displayed, one of your filters, or the entire x9 file. Images are formatted and inserted into a PDF template which you can customize. You\ can control the number of checks that are printed per page, paper size, orientation, and the background template that is used for printing.

Tool	Usage
Validate	Run all x9 validations. This function would typically be used after modifications have been made to the x9 file.
Scrub	Scrub the file to remove all proprietary information. The types of proprietary information that will be removed is selectable as part of the scrub operation. This function is very helpful to allow you to create and then ultimately save a new x9 file that is based on this x9 file but has customer or operational information removed based on confidentiality requirements. For example, you can use scrub to create a new x9 file that has replaced customer account numbers and the check images with randomly created information.
Modify	Modify allows you to start or stop a modify session, as well obtain a list of all changes that have been made to the current x9 file. Modification includes the ability to modify individual fields, delete certain x9 record types, and replace images with missing image documents. Modify allows you to review a list of your field changes and then easily jump to any record that has been changed. Modify also provides a facility to revert any individual change that has bee made.
Delete	Delete the currently displayed x9 record when a modify session is active.
Repair	Apply automated repairs against the currently loaded x9 file. Repair will provide a list of all currently implemented repairs that can be applied against an x9 file. You can then select from this list and initiate the repair operation. A new file validation will be run once the repairs have been applied. You can then review the results and determine if they meet your specific needs. The repaired x9 file can be further modified and then ultimately saved based on your requirements.
Import	Import and internally create a new x9 file from an external Excel file (XLS or CSV). Please refer to the import documentation for more information on this functionality.
Export	Export the current x9 file to an external CSV file and optional write the images to an output folder of your choosing. This CSV file can then be used for other purposes or routed to other applications for processing. It is also possible to make changes to the resulting CSV file and then import those results back into X9Assist. Please refer to the export documentation for more information on this functionality.

Tool	Usage
ItemViewer	The Item Viewer provides full scale images for easy viewing and allows you to page back and forward to view checks within the current bundle. The Item Viewer allows you to flip a check from front to back. It also provides functionality to export an image to an external file and copy the image to the system clip board.
Make	Make is used to dynamically create X9 files from your specifications. Input can come from either CSV files or Excel worksheets Make allows your to define reformatters which translate your use cases to the input required by Generate.
Generate	Generate is used to dynamically create X9 files from your specifications. Generate will dynamically create images to match your x9 items. Generate also wraps all of the generated items with the control records and totals needed for a completely valid x9 file. The generated x9 files can be used by your internal application testing to resolve many confidentiality issues associated with the use of production files in your test environments.
Clone	Clone is used to create large x9 files by cloning (copying) the bundles within an existing x9 file.
Merge	Merge allows you to merge the contents of another x9 file into the currently loaded x9 file. Merge is a convenient way to combine multiple x9 files into a single (larger) x9 file. This is helpful to combine various x9 files with different test cases into a single file that can be used for ongoing testing, or to combine various x9 files to create a single larger file, possibly to be used for stress testing purposes.
Image MisMatch	Image MisMatch allows you to create checks where the front images are mismatched with the associated MICR line data (either ICL type 25 or ICLR type 31). This is a helpful process when you require image mismatch files for your downstream application testing.
Duplicate Items Detector	The Duplicate Items Detector can identify duplicate items either within the current x9 file, or across multiple x9 files. Duplicates can be identified for the following types of items: check detail records (type 25), return detail records (type 31), and credit reconciliation records (type 61). Duplicates are considered within record type, which means (for example) that you cannot have a duplicate that consists of a debit and a

Tool	Usage
	credit, even if they have the same MICR line and amount.
Add Item to Picklist	Add the currently positioned item or bundle to the Picklist. You can then use the Picklist for other functions such as creating an ICL/ICLR file or deleting the selected items.
Delete Items using a Filter	Delete the items from the current x9 file using items that are enclosed in a filter or the Picklist. You can select these items in one of several ways. Filters can be created using either the Find or Search tools. Individual items or bundles can be added to the system Picklist.
Generate ICL/ICLR	Create a new ICL/ICLR file using the items from the currently loaded x9 file, or from the items that are enclosed in a filter or the Picklist. You can select these items in one of several ways. Filters can be created using either the Find or Search tools. Individual items or bundles can be added to the system Picklist.
Merge Filters	Merge two or more filters. You can then use the merged filter for other functions such as Delete Items or Generate ICL/ICLR.
Release All Filters	Release all currently defined filters.
Mapping Editor	The Mapping Editor allows you review and update your mapping rules, which are used to dynamically “bind” to configuration rules based on contents of the x9 file header. This process allows the correct rules to be applied to the file on an automated basis without manual intervention. For example, you can use the bind process to define those exchange partners who provide x9.100-187 files versus those who provide x9.37 files, when fields in the x9 file header are not assigned properly to allow that detection to be done based on the file alone. X9Assist is then able to apply the appropriate validation rules based on the originator.
Configuration Editor	The Configuration Editor allows you to define logical configurations that are used for x9 and tiff validation. Each configuration consists of a x9 rules file, a tiff rules file, an application options file, and an error message severities file. Each of these is an xml file which controls part of the file validation process. A combination of these four files can then be defined as a logical configuration and given a name which is used by the Mapping Editor for x9 file binding.

Tool	Usage
ABA Editor	The ABA Editor is an optional facility exists to define your upstream and downstream partners by their ABA. You can then indicate if their role (upstream, downstream, both) and their status (start date, end date, etc). This allows X9Assist to validate the origination and destination ABA's that exist in the file header. This ABA file can be keyed or extracted from your image exchange application.
ABA List Editor	The ABA list editor is used to create ABA lists that can be used by several functions including the Scrub and the Use Case Editor. ABA numbers are created with the appropriate MOD check digits in FED districts of your choosing.
ModCheck Editor	The Modcheck Editor is used to maintain a list of available account number modulus routine definitions, which are input to the Use Case Creator. This list allows account numbers to be created that will appropriately self check for a given financial institution.
Use Case Editor	The Use Case Editor will generate use case lists that can then be used by Make and Scrub. A use case list contains information that is used to populate type 25 or type 311 detail records. This includes the aba, account number, process control, and AUX OnUs fields which are all required by the MICR line.
RT Calculator	The RT calculator is a simple tool to calculate RT check digits.
System Log	The system log is used when researching issues either with X9Assist or your x9 data. Use of the system log can be complex and should be used when you are working directly with X9Ware.
Program Options	Program Options are used to display and optionally make changes to the default settings. You can update the current program options or create a new options file. All option files must be maintained within the xml/options folder. Program options are configured during X9Assist program startup, so you will have to restart the application to see the impact of your option changes.
License Key Editor	The License Key Editor is used to enter a new X9Assist license key or review the current license key that is in place.
Help	Help will display a list of the available program

Tool	Usage
	documentation. You will then be able to view and search those individual documents to learn more about X9Assist.
About	About provides detailed information about the X9Assist application and X9Ware as a company that wants to meet your x9 needs. There is also a link to our web site. If you have any questions regarding X9Assist, licensing options, enhancements that you would like to be made to the product, or possibly consulting opportunities, you can email us directly at x9assist@x9ware.com.
Exit	Exit the application.

Opening a New File

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

New files can be opened in one of several manners:

- Using the open button on the tool bar;
- Using the open recent button on the tool bar;
- Selecting FILE and then OPEN from the menu bar.

Files are automatically identified (x9, ach, cpa005, etc) based on their file name and contents based on the following strategy:

- File extension matching is applied first, where Program Options contains a list of the file extensions for each of these categories. This assignment (using the file extension) takes precedence over subsequent methods.
- Data inspection is applied second, where record content and data lengths are used to explicitly identify x9 versus ach data files. This logic leverages optional binary record lengths and then specifically inspects for valid record types subject to format.
- Invalid files (those that fail all inspection techniques) are assumed to be in the x9 format and will ultimately be displayed as structurally flawed.

You can use Program Options to simplify the process of opening new files:

1. The default input file folder location.
2. Wildcard strings which are used to explicitly identify the format of user file names. These wildcard strings can be as simple as simple as *.x9 or *.ach, and thus identify files based on file name extensions, or can be more complex using file name patterns and the wildcard characters * and ?. Within the wildcard patten strings, the special character (?) Is used to represent a single wildcard character while (*) is used to represent one or more characters. For example, a more complex wildcard string might be *icl*.x9 which would accept file names with the string “icl” embedded within their name and with a specific file extension.

By default, the list of files is restricted to those files that have file names that match the wildcard strings which are assigned in program options.

Please remember that a list of file extensions (or more complex wildcard strings) are defined in program options and can be customized to simplify your file selection process. You can review and update this list of file extensions by invoking the options maintenance facility. These file extensions are maintained on the primary “x9” tab.

The “**Look in**” box at the top of the file open dialog can be used to easily move to any folder location that you need across the entire file system. Using this box, you can navigate to the drive

and folder that you want to select from You can then select and open the image cash letter you would like to open.

If you need to open a file with some other extension, you can use the "**Files of Type**" drop down at the bottom of the panel to change to All Files. This will display all files in the current folder regardless of extension, which will allow you to select any file with any extension. You can also use "**Files of Type**" to individually select your custom wildcard selection strings.

As a reminder, HELP is provided on the right side of the file open panel. This help information serves as a reminder of how to use the file chooser panel. You can HIDE this information once you are familiar with the process.

Panel Dimensions

The file chooser panel can be resized (both horizontially and vertically) to increase the number of files that can be viewed based on your monitor size. If you resize this panel, your new dimensions will be saved in preferences and applied to future sessions.

List versus Details Mode

You can use the View Menu drop down box in the upper right hand corner to select either list or details mode. Details mode provides access to both file dates and file sizes which can be very helpful when searching and selecting files. File chooser columns can be dynamically sorted on these columns. View mode can be defined in program options and your most recent setting will be saved in preferences and applied to future sessions.

Supported x9 File Formats

X9Assist supports all file formats that are currently exchanged between financial institutions, and will automatically detect the attributes of the file being opened. Once these attributes are determined, they will then be checked against your rules to determine if attributes (as present) are valid or should be considered as an error. These file attributes are as follows:

Character set	The character set can be Ebcdic or Ascii. This is determined by inspecting the first record on the file (which will be the file header) and assessing the character set based on the how the record type is encoded within that record. The UCD standard is Ebcdic character set encoding.
Field Zero	The term "Field Zero" is often used to describe the four byte record lengths that may exist as a prefix to each x9 record. X9Assist can run either with or without these record lengths. If the field zero lengths are present, X9Assist can process then in either big endian or little endian byte order. The UCD standard is for field zero to be present and as big endian.

Tiff Images	Tiff images include an Image File Directory (IFD) which can be defined in either big endian (Motorola) or little endian (Intel) byte order. The UCD standard is for Tiff images to be in little endian byte order.

File Chooser Dialogs

File dialog boxes are used through the application to open and save files. This functionality is used extensively within X9Assist so it is important to leverage the capabilities that are available to you.

X9Assist will always suggest the initial folder that is used by a file dialog box to select a file. This will be helpful in some situations, depending on the location of the last file that you opened and what you intend to do next. This functions as follows:

- The last input path is remembered and used as the starting folder for open input requests.
- The last output path is remembered and used as the starting folder for open output requests.
- There is an override in program options that indicates open output operations should be alternatively routed to the last input path. By default, this option is enabled, which allows you to easily save output files in the same folder where they were opened from.

Program Options allows you to select one of two file dialog interfaces:

- A Java based file chooser, which is unique to Java and is independent of the current operating system. This is the default option.
- A native OS file dialog, which uses standard operating system facilities in support of file selection interactions. For example, in a Windows environment this will be the standard Windows file dialog that is typically used by most applications.

There are pros and cons associated with both of these file dialog interfaces. We support both since one or the other may be more desirable depending on your usage and your specific environment.

- Extensive testing has shown that the native OS file dialog tends to be more robust and performs well when your directories contain a very large number of files. However, with the introduction of Java SE7, the Java file chooser is performing better, so this appears to be less of an issue. The Java file chooser had performance issues when folders contained large ZIP files, but that issue was also resolved with SE7.
- Unfortunately, the native OS file dialog does not support file filtering and does not provide any easy mechanism to filter based on file extension. We have an implementation for that, but is less than desirable. The Java file chooser does a much better job in this area.
- The native OS file dialog includes some additional file functions that are not available with the Java file chooser. These options are available by right clicking on the file name within the dialog box. For example, you can easily rename a file within the dialog.
- Our implementations allow you to resize either dialog box and we will persist those custom dimensions in the preferences file. The resized dialog box will be used in future sessions.

Bookmarks

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

File choosers are used throughout X9Assist to select input files and folders for a myriad of functions. X9Assist has two flavors of file choosers that can be utilized:

- A File Chooser, which is based on internal Java capabilities.
- A File Dialog, which is based on the underlying operating system (eg, Windows).

The default is File Chooser, which can selected on the “Gui” tab within program options. These program options can be viewed and updated using / Help / Options / on the toolbar.

When File Choosers are being utilized, the bookmark function can then be used to record commonly used directories (folders). Using bookmarks can be a time saver, since they eliminate the constant lookup of folders within the file system. Once you locate a folder, you can bookmark it, allowing you to jump directly to that folder in the future.

Creating Bookmark Files

Bookmarks are stored in folder / x9_assist / xml / bookmarks /. Initially, there is a single (global) bookmark file that is used throughout X9Assist. However, you can create additional bookmark files for specific functions and then use those specific bookmark references when needed. For example, you might create separate bookmarks for production support versus development, or you might create a separate bookmark file for each major project you are working on. Bookmark files are selected using the drop-down box. The “create” function is used to create a new bookmark file, while the “delete” function is used to delete a bookmark file. Both of these include popup confirmation messages.

Maintaining Bookmarks

There are several especially important features which make bookmarks even easier to use:

- The bookmark list is automatically saved as you make changes to them. This means that you do not have to go through the work to save them; that just happens behind the scenes.
- The bookmark list is self maintaining; entries will be automatically removed when the underling bookmarked folder no longer exists.
- Bookmarks by default are assigned names based on based folder name.
- Bookmarks can be renamed using the “=” icon on the task bar.
- Bookmarks can be deleted should they no longer be needed.

Dashboard

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

The Dashboard contains high level information regarding the currently loaded file that has been parsed and validated. It serves as the summary panel for all information that has been collected and allows you to easily launch various system functions.

The Menu Bar is available at the very top of the panel and is used to launch application functions. Each entry within the menu has a drop down that is activated on a single click. Most functions have an assigned keyboard speed that can alternatively be used for initiation.

The Tool Bar (which contains icons) is available immediately below the Menu Bar and can similarly be used to launch application functions. The Tool Bar can be easier to use than the Menu Bar for commonly used tools.

The Dashboard contains tabs on both the top and right sides:

- Tabs with their title in **black** contain information at the file level. These tabs are presented in right side tabs within the Dashboard. For example, the “Records” tab contains summary information about the various record types that exist within the current file.
- Tabs with their title in **blue** contain detailed information which will be updated as your positioning within the file changes. For example, the tab for the “Field Viewer” has field level information for the current record group that is selected within the browser tree.

The Dashboard top side tabs are as follows:

Dashboard	A summary of information associated with the file that is currently loaded. This includes bulletins (messages) and a summary of all identified errors.
Field Viewer	A window shows the current record or record group, on a logical record by record and field by field basis.
Images	A panel that allows selection of either the front side or back side image (x9 files only).

The Dashboard right side tabs are as follows:

Tab	Applies to X9?	Applies to ACH?	Description
Summary	Yes	Yes	A summary of information associated with the file that was processed. This includes file attributes and the record types that are present on the file.
Info	Yes	Yes	A list of the file level bulletins that were created when the

			file was loaded and validated.
Records	Yes	Yes	A summary of all record types that exist within the file.
Errors	Yes	Yes	A detailed list of the errors that were found within the file.
Cash Letters	Yes	No	A list of the cash letters that exist within the file.
Bundles	Yes	No	A list of x9 bundles that exist within the file.
Batches	No	Yes	A list of ach batches that exist within the file.
Items	Yes	Yes	A list of the items that exist within the file.
Tiff Tags	Yes	No	A summary of all tiff tags across all images within the file.

The feedback panel is displayed at the lower portion of the Dashboard. It contains Record Data, Errors, and the Images panel. It is located at the bottom of the Dashboard and presents additional information for the current record or record group that is selected within the viewer.

Record Data	Shows the complete record data exactly as it was received. This can be very helpful when there are questions about the positioning of fields within the record or how fields are justified or padded. All errors are listed after each individual record.
Thumbnail Images (x9 files only)	The front and back image thumbnails are shown in the lower right corner of the Dashboard. This is helpful to allow you to quickly see both the front and back images for each check. The Item Viewer is launched when either thumbnail is clicked, providing a larger image view. Once the Item Viewer is displayed, you can then click either the front side or back side image for an even larger view of these images. The size of the thumbnails can be changed using sliders that are available on both the left and top of this image area. These sizes are recorded in user preferences and will be used for each user session.

The ICL Viewer tree is on the left side of the panel and can be used to view the structure of the overall file and to launch the display of any given record. The tree includes a tab which contains navigational information which describes how the tree is used and speed keys that can be used for simplified access.

The Status Bar is displayed at the bottom of the Dashboard and shows attributes of the current file. This includes the file name, file type (x9 or ach), rules configuration that was used for validation, record count, file size, character set encoding, and finally the number of errors that were identified during validation (errors, warnings, infos).

The Progress Bar shows the percentage of work that has been completed for a given task. The Progress Bar will go to 100% and will then disappear when a task has been completed.

Item Viewer

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

The Item Viewer is launched from the DashBoard using either the tool bar or the menu bar and provides access to several panels and tools that can be leveraged for the current item. The purpose of the Item Viewer to to use the entire panel to provide information regarding the current item. For x9 files, a large scale image can be displayed which provides image details that otherwise may not be readily visible from the smaller images that are displayed within the DashBoard.

The Item Viewer allows the current item to be displayed in one of several modes which are selected from the action line at the bottom of the panel. The available modes are:

Tab Name	Content	Applies to X9 ?	Applies to Ach?
Viewer	A split panel shows the fields for the current item on the left, and the front/back images on the right (for x9 files).	Yes	Yes
Front	An image panel shows the item front image (for x9 files).	No	No
Back	An image panel shows the item back image (for x9 files).	No	No
Tiff Tags	A split panel shows the current image on the top and a table of the associated TIFF tags on the bottom (for x9 files).	No	No
Hex	A hex panel shows the current item record group in hexadecimal format, which includes the item and all attached types within the item group.	Yes	Yes

Viewer Panel

The Viewer panel is a consolidated panel which contains detailed information for the current item, which includes the field number, field name, field value, and an icon which represents any errors which were identified during validation

Field Values with Leading/Trailing Blanks

The value column is formatted to provide insight into how and where blanks are present within the data value. Leading and trailing blanks are displayed as a shaded gray area to make their presence and visibility more apparent. This formatting does not impact embedded blanks, but only the leading and/or trailing blanks that may appear as part of field justification.

Browsing Functions

The Item Viewer includes previous and next functions which allow you to easily walk through the items that are located within the current file, using the entire viewing space of your screen size, thus maximizing the data you can see and scroll through. When paging forward or backward, the Item Viewer will continue to show items until the very last item within the current file is displayed. This browsing process may automatically move forward to the next bundle or cash letter based on the current file structure. The Item Viewer will return to the DashBoard when there are no more items to be displayed within the current file.

Modifying Images

When users are licensed for modify, the Item Viewer has tools which allow the front and/or back images for the current item to be modified. This is accomplished using various icon-based tools that are available on the toolbar, followed by a replace function. Our UI has implemented image modification here, since this allows you to actually see the image as changes are made. For example, possible steps to modify an image might be:

- Sharpen the image.
- Deskew the image for better horizontal alignment.
- Repair the image to rescale (if needed) and recreate tiff tags.
- Replace the image, which attaches the modified image to the current item.
- Closely review results.
- Save the modified file to an alternate file name.
- Use X9Assist-Compare to compare the original file to the modified file.

Once the replace function is used to attach a modified image to the current item, it can be subsequently reverted from the modification log.

Item Viewer Tools

The following x9.37 tools are available within the Item Viewer:

X9.37 Tool	Description
Export	For x9 files, individual images can be exported to externally stored image files. These images can be written in TIFF format, but can also be written in other standard industry image formats such as PNG, JPG, GIF, etc. When images are exported in TIFF format, the exported TIFF file will contain the ***exact*** TIFF image data that exists on the x9 file (the same segments, TIFF tags, etc).
Copy	For x9 files, individual images can be copied to the clipboard. This is a very convenient way to move images from X9Assist to other desktop applications.
Strip	For x9 files, appends a MICR strip using the actual item data from the current item record. The data will be extracted from the current item record which may for example be a 25, 31, 61, or 62. This function is very useful when you want

X9.37 Tool	Description
	to visually compare the image against the data that exists within the file itself.
OCR	For x9 files, appends a MICR strip using data from an OCR Read, which is based on the X9Ware-E13B-OCR product. The MICR data is extracted from the actual front side TIFF image. This function is very useful when you want to visually compare the image against our OCR reat results. The ? Button can subsequently be used to gain more insight into our E13B recognition process.
MICR Gauge	For x9 files, displays a MICR gauge as an overlay against front side images. This tool allows you to easily confirm that a MICR line has been properly positioned within the clear band area (the bottom 5/8 inches of the image).
Density	For x9 files, displays the pixel density for the current image, which is either the “top” image (when both images are being displayed) or the current when in zoom ode. This tool can be used even when IQA is not enabled and provides insight into the too light or too dark attributes for any given image within an x9 file.
Sharpen	For x9 files, applies our sharpening tool to the current image and thus attempt to remove background noise. Depending on the condition of the image, the sharpened image may improve your image visibility and otherwise bring data into focus which is otherwise not readily viewable.
Blur	For x9 files, applies our standard blurring tool to the current image to simulate scanner issues. If our standard blur does not meet your needs, we have included steps (documented below) to export an image so you can apply blurring using an external image tool such as GIMP.
Deskew Left	Slightly rotates the image left to attempt to remove possible capture skew. The undo button can be used if the results are not acceptable.
Deskew Right	Slightly rotates the image right to attempt to remove possible capture skew. The undo button can be used if the results are not acceptable.
Flip	Changes the image orientation by 180% (horizontal flip).
Exchange	Exchanges the front and back images for this item.
Repair	Attempts to repair the image using our TIFF repair capabilities.
Undo	Undo recent changes that have been done within this item viewer. Undo cannot reverse changes after a replace has been done; that can only be done through the modification log.
Next	The next item will be displayed within the current file.
Previous	The previous item will be displayed within the current file.

X9 Image Replacement

Image changes are logged in the same manner as field level modifications. Image changes can similarly be viewed using the modification log and can be reverted in the same manner.

Images can be replaced using one of the following standard facilities:

- **Current viewer image:** the current viewer image will be used to construct the replacement image. This image may have been modified using one or more tools such as sharpen, deskew, or repair.
- **Load external image**: a replacement image is loaded from an external image file. If the replacement image is in TIFF format, then the byte array for that image will be inserted into the type 52 record exactly as it appears in the external image file; no image manipulations will be performed. If the image is in another supported format (PNG, JPG, GIF, or BMP) then it will be converted to a standard TIFF image that is compliant with the x9.100-181 exchange standard. This includes conversion to black/white as needed. You retain the most control over your replacement image when it is provided in TIFF format, hence it is our recommendation to do that whenever possible.
- **Missing**: the standard missing image document will be used to replace the current image (which might be either a front side image or a back side image). The missing image document is a standard document that can be customized to contain specific data per your site requirements. There are separate missing-front and missing-back images that can be defined as needed.
- **Both as missing**: the standard missing image documents will be used to replace bot the front side and back side images for the current item. This is typically done when the wrong item image is associated with the current x9 item.
- **Zero length image**: field 52.18 will be set to zero which implies that no image is present within the type 52 record. However, the type 52 record is retained within the item group.

Blurring an Image

It is occasional desirable to blur an existing image, as a negative use case for image processing. Creating blurred images is a complex process for several reasons:

- The image manipulations needed to blur an image are in themselves complicated.
- There are various blurring techniques that can be applied which will result in different results. These various tools need to be tried through a trial-and-error process, to achieve the best output subject to your requirements.
- Various parts of an image can be blurred. You may want to blur parts of the MICR line, the CAR/LAR fields, other image fields, or the entire image to emulate a bad camera.

The toolbar includes our standard blur which can be applied to an image. However, if our standard blur does not meet your needs, you can use this alternative sequence of tasks to utilize an external tool (like GIMP) to manipulate the image. GIMP not only has various blurring tools, but also allows the blurring to be applied selectively to the image, with do-undo support as you work on the image. You do not need to use GIMP, you can use another tool of your choice.

The sequence of steps to accomplish image blurring (or other image manipulations) is as follows:

- Go to the item that you would like to manipulate.

- Double click the front thumbnail image, which will put you into the item viewer.
- Using the buttons at the bottom, you can export this one image out to a file name of your choice.
- Now use another image tool to blur this image. My tool of choice would be GIMP. Once you've finished, export this image as a PNG.
- Go back to X9Assist, open up the file, and go back to this item.
- Turn on modify using the upper right hand corner check box that's on the dashboard.
- Now modify the type 52 front image record.
- Inside the modify record panel, you will see field 52.19 at the bottom. Click modify to modify this image.
- Use the button options to indicate you would like to replace the image from an external image.
- Go into the file system, and select the image that you have blurred and saved.
- The selected image will now replace the original image, and you will see it in the item viewer.
- Finally, you can say this X9.37 file and you will have the image exactly as it has been modified and within this file.

Now that you have the modified file that contains a blurred image, you can either send it for image processing, to a downstream application, or print if that would be preferred.

Hex Viewer

The Hex Viewer allows individual records to be viewed in hexadecimal format, as they appear on the input file. This viewer provides several important capabilities:

- Data is logically parsed into individual records based on actual data length
- The four byte record length prefixes are shown when they exist within the file (when they exist as the physical length of each logical record)
- For x9 files, the tiff image data from the type 52 image view data records is included
- Each data record includes:
  - Record number
  - Record length
  - File offset
  - Hexadecimal data (in either the Ebcdic or Ascii character set)
  - Translated character data

Structurally Flawed Files

An important feature of the Hex Viewer is that it is automatically invoked when a structurally flawed file is opened. The purpose of this interface is to provide as much information as possible when flawed files are opened.

These are always difficult situations and can easily turn into complex discussions. Most x9 tools provide little to no information on flawed files. However, X9Assist is designed to intercept these parsing errors and give you as much insight (as possible) as to where the error exists.

When a structurally flawed file is open, X9Assist will:

- Provide a pop-up message with information on where the problem was encountered.
- Will then position you within the Hex Viewer at the point of the parsing failure.

The pop-up message contains the following information:

- The number of records parsed up to the break point
- The percentage of overall data that was successfully parsed

Validation

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

X9Assist is designed to apply all applicable validations to the incoming x9 and ach files. Each record field can have one or more assigned validations, all of which must be satisfied for the field to be considered to be valid. The field viewer shows the validation rules which are being applied to each field which is helpful to gain insight when running the same file against alternative rule sets.

X9Assist is supplied with an extensive list of configurations which implement the validation rules for each of the commonly used standards. These validations are implemented through system supplied XML configuration files. It is possible to extend the standard edits that are packaged and distributed, but also please be aware that this is an advanced topic which requires a thorough understanding of the standards themselves. It is recommended that any user modifications be stored in separate configurations so they do not overlay the standard rule definitions. As part of Extended Support, X9Ware will provide consulting services to help with any such customization efforts.

There are various field level edits that are defined within the validation standards:

- Field presence can be either mandatory or conditional. The standards state that a mandatory with a defined values list must contain one of those values to be considered as valid. Validation of conditional fields is more complex, which the term conditional means that the content of one field may be conditional on the content of other records and fields within the currently loaded file. Generally, conditional fields are essentially optional where their content can be blank, which means that a conditional field with a defined values list can normally be blank and still considered to be valid. However, this validation is dependent on the specific conditional field and the rules surrounding that conditional presence. The standard developers are a bit vague in these areas, with the presence of many fields identified as based on “clearing arrangements” which indicates that the rules are agreed upon by the originator and receiver.
- Fields can be defined with a data format attribute. For example a numeric field can only contain the characters zero through nine, which a numeric blank field can contain the values zero through nine as well as trailing blanks.
- Fields can be assigned a data alignment attribute, which indicates if padding blanks (when present) must be on the left or right. For example a numeric blank field is justified left which dictates that any padding blanks must be on the right.
- Fields can have a defined values list, which provides a specific list of allowable values for the fields. A field can also have multiple value lists, where its allowable content varies based on the value of other fields that exist within the file.

Rule Definitions

File validation is driven by xml documents which define record types, fields, field attributes, and the edits that are applied to each field. These definitions are located in the following folders:

- /rules/messages/: contains a list of error messages and the severity assigned to each. Although most installations will have a single messages file, it is possible to implement to vary the messages (with their severities) based on the file standard.
- /rules/tables/: contains a series tables with a list of values by table name. Table validations are applied by the <edit>table/tablename</edit> rule. Tables are useful when the same list of values are applied to numerous fields and there is a desire to only define those allowable values in a single location. Tables are also useful when the list of valid values is extremely lengthy and thus not a good candidate for direct value checking.
- /rules/tests/: contains a series of test sets which are designed as test-pass-fail rules. Test set validations are applied by the <edit>/test/testSetName</edit> rule. Each test set consists of a series of tests which are applied sequentially to validate a field value. Each test results in true or false. A test set is considered to be passed when the first successful test is encountered. If all tests are false, then the field value has been determined to be unacceptable and failed. Each test test assigns its own unique error message to describe the failure, along with an optional severity level. Refer to the SDK User Guide for more detailed information regarding the test set rules facility.
- /rules/tiffrules/: contains a series of tiff validation rules which define the allowed attributes of the tiff images that are embedded within the file. The tiff rules contain parameters to allow the values of individual tiff tags to be validated, and also indicate which tiff tags are mandatory and must be present within the tiff images. The tiff rules can be used to dictate the image format of the primary binary (bi-level) images that are attached to each item, as well as optional secondary images which can be gray scale and thus contain a different set of attributes and tiff tag requirements. Refer to the SDK User Guide for more detailed information regarding the TIFF validation rules.
- /rules/x9rules/: contains a series of validation rules which define the allowed content of files. The rules represent the core of our validation engine and drive the validation of each file. These rules indicate which record types may be present within the file, the fields that make up each record definition, and the validations that are applied to each field. Refer to the SDK User Guide for more detailed information regarding the validation rules.

Validation Rules

The rules engine implements the following proprietary design principles:

- A validation set can be either a basis document or an extension document.
- A basis document defines a core specification.
- An extension document defines a new specification that is built upon a defined base. When defining an extension, only the differences (against the base document) need to be defined and all else will be automatically inherited. This approach substantially reduces ongoing maintenance, since a change basis is automatically applied to any defined extensions. It also allows you to quickly and easily understand what is different in the extension specifications.

- Each specification has a set of controls which define the high level attributes associated with the rules. For example, the controls might indicate if the allowable character set is EBCDIC, ASCII, or EITHER.
- A base document must contain a controls definition and need only define those values that vary from the system defaults.
- An extension document must contain a controls definition and need only define those values that vary from the base document.
- Each specification is defined as consisting of a series of record types.
- Each record is defined as consisting of a series of fields.
- Each field is defined with its specific attributes and validation criteria.

TIFF Rules

The TIFF rules engine similarly implements the following proprietary design principles:

- Each TIFF specification has a set of TIFF controls which define the high level attributes associated with the TIFF rules. For example, the TIFF controls might indicate that duplicate tags are accepted or are not accepted.
- Each TIFF specification can have separate set of TIFF rules for Black White versus Gray Scale images.
- Each set of TIFF rules contains a list of TIFF edits that are applied to tags when they are present within an image.
- Each set of TIFF rules contains a list of TIFF tags that are mandatory for each image.
- The TIFF specification contains a list of descriptions of all possible tags that can be present across all images within the x9 file.

More Information...

Please contact X9Ware for more information regarding these validation rules and how they can be extended to meet your specific requirements.

Image Quality

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**NO**	**NO**

Image Quality tests can be performed during file validation processing. These functions are enabled from the tool bar and must be set prior to a file being loaded. This design minimizes the impact of performing these tests since they share the same file read operation as is used by validation. The image quality tests that are available are as follows:

- IQA (Image Quality Analysis) which will check if images are too light or too dark
- ECI (Electronically Created Image) Detection which is used to identify suspect items that were computer created and were not physically scanned.

Image Quality Analysis

In the image capture environment, IQA can encompass more sophisticated tests than our more simplistic too light and too dark tests (for example, streaks, bands, edge detection, piggyback detection, and so forth). These more complex IQA functions are currently beyond the scope of what has been currently implemented within our current inspection tools. There are several reasons behind our current implementation and capabilities:

- Too light and too dark are the minimal IQA tests performed on a standard basis by many processors or financial institutions. There are published industry standards around IQA settings (for example, the FRB has published their too light / too dark settings on their website.
- Too light and too dark tests can be performed with a minimal amount of system resources by computing pixel density. This makes the tests faster than the more complex tests. Our image inspection process has implemented a multi-threaded approach that results in a high performance solution that produces results in a minimal elapsed time.
- Although the too light / too dark tests are simplistic, they are very effective in identifying poor images. This makes them very popular by most image capture systems.
- Too light / too dark are useful tests which can catch a variety of image related errors. For example, they often detect camera issues, streaks, bands, and swapped front-back images.
- Too light / too dark are obviously the easiest IQA tests to implement. Our current design is to not perform as a full fledged IQA product, but to instead provide meaningful information on a timely basis.

On a default basis, image inspection will tag all IQA failures as informational errors. You can change the severity of these errors by updating the appropriate messages xml configuration file.

On a default basis, image inspection applies the too light and too dark tests to the front image only. Examination of just the front will allow detection of typical camera issues as well as swapped front versus back issues. The front issue is also the most critical from an image exchange perspective.

This is a clearly a time tradeoff, since examining the back images will typically double the amount of time required to perform these tests.

Errors are thrown during validation for items that do not meet the defined too light or too dark thresholds. When an item fails a too light or too dark test, the actual pixel density can be obtained in one of two manners:

- Use the record viewer (in the lower left of the dash board) to interrogate the density that was computed. The pixel density is calculated as the number of black pixels as a percentage of the total pixels that are present. You will have to slide the record viewer to the right to be able to see the supplemental information for these errors.

- Use the Item Viewer to examine the item, which can be launched either using Tools / Item Viewer or from the menu bar. You will then have to activate pixel density using the check box at the bottom of this panel.

Electronically Created Item Detection

ECI is used to identify suspects items that are believed to be computer (electronically) created. ECI detection may be an important consideration (especially during the onboarding of new customers) given the potential fraud associated with these items. Our inspection process implements imaging techniques which look for “perfect” images, which are typically achieved when the image is computer drawn but is otherwise not possibly achieved when images are physically captured. ECI images are identified using two separate algorithms:

- Pixel noise per square inch, which is separately associated with the front and back images. Random pixel noise, especially around captured text and graphics, is typically introduced by the capture process but would not be drawn into an electronically created image.

- Horizontal image skew, which is calculated from the front side image and represents any slant that may exist within the capture image. Computer drawn images do not have any skew and are very horizontally level from left to right.

ECI detection can be used identify “remotely created checks” which represent an electronic authorization to debit an account. In this transaction scenario, the image itself is computer drawn (as proxy for a customer check) and does not bear the signature of the the person on whose behalf the check is drawn. The RCC will instead include the account name or perhaps just a statement that the account holder has authorized the payment. The RCC must include a valid MICR line for the item which includes the bank routing and customer account number. RCCs are often created by credit card companies, utilities, and telemarketing operations. RCCs are processed through the x9.37 check clearing network and may be alternatively be converted to an ACH debit for clearing.

Activation

IQA can be activated in one of two manners:

- IQA can be activated on a default basis using the tiff rules that are assigned to the current configuration. For example, you can define certain configurations to automatically perform IQA while others do not.

- IQA and/or ECI can be selectively activated using the tool bar. Once activated, the selected function will be applied to the next file that is loaded and validated.

Settings

IQA/ECI settings can be established as follows:

- Default IQA settings are specified through the tiffRule xml configuration file. Note that you can define various tiff rules that can be automatically applied through your configurations. These parameters allow you to activate IQA and then set their pixel density thresholds.

- The tiff rules XML parameters are defined on a default basis from the FRB web site, as follows:

```
<applyIqaRules>false</applyIqaRules>
<applyIqaToFrontOnly>true</applyIqaToFrontOnly>
<frontTooLight>0.900</frontTooLight>
<frontTooDark>90.000</frontTooDark>
<backTooLight>0.380</backTooLight>
<backTooDark>98.000</backTooDark>
```

- File / IQA Settings / can be used from the menu bar to view and optionally modify the IQA settings for the current user session. These settings are not saved across user sessions.

Summary

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

The Summary tab provides high level information about each of the record types that is present on the currently loaded file. There is a summary line within the display panel for each record type and record format which exists within the file. Each summary line includes:

- Record type
- Record format
- Record description
- Record count (the number of these records that are present within the file)
- Total amount as summarized by this record type and format
- Number of associated errors
- Number of associated warnings
- Number of associated informational messages
- An ICON which shows the overall error status for these records

The Summary tab allows you to quickly see high level information regarding the current x9 file. For example, you can tell if the file contains credits. In fact, in the case of type 61 credit reconciliation records, you can determine the additional information as to which credit record format is present. You can also review the totals and determine the overall balance status across all record types.

The Summary tab is rebuilt whenever a file is validated, since the application of a different set of rules may result in changes to the error counters. You can reference the Errors tab to see the detail behind any of the reported errors.

Error Summary

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

The Error Summary appears within the dashboard and provides a summary of all errors that were encountered during validation of the currently loaded file. This validation is based on the rules configuration that has been applied, either on an automated (auto) or manual (set) basis.

It is important to note that the indicated errors may vary depending on which configuration is used to validate a given file. Because of that, it is sometimes advantageous to validate a file with several different configurations to better understand the content and format of the file.

Each error summary line includes:

- Error Field which identifies the record/field number associated with this error
- Field name
- Severity icon associated with the error per the message rules (error, warn, or info)
- Count of errors of this type
- Description of the error
- VIEW button which can be used to launch more information for this entry

Errors are logically sorted from the highest occurrence to the lowest occurrence for the entire validation process. This allows you to quickly see the reasons behind the errors that were found within this file. You can also easily see the errors that were created with the highest frequency.

You can refer to the Errors tab to see the detail behind all of the errors that are reported. You can also launch directly to any error record using the Errors tab.

Errors

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

The errors tab provides a summarized list of all conditions that were identified during validation of the current file (x9 or ach) and are summarized by error description. The number of errors of each type are shown, which allows you to easily review the types of validation errors which have been identified within the currently loaded file.

Individual entries can be expanded or collapsed using the first column within the table, allowing you to focus on specific error conditions. You can expand and collapse each error type using the first column within the table. Once an error type is expanded, you can use the VIEW column button to launch to the specific record that is associated with this error. This will allow you to get more information about the record, field, and other context associated with the specific error.

The Error tab includes horizontal scrolling which allows you to view all of the fields that are present for each error message. The fields presented on the Error tab will also appear in this same order within the data created by the Excel Exporter. This information can also be directly (and selectively) exported directly using the Export function. The columns provided within the Error tab are as follows:

1) Error description
2) Error identifier
3) Error record number
4) Error field number
5) Error field name
6) Error field data position
7) Error field data length
8) Error field value
9) Error field mandatory indicator
10) Error field list of allowable values
11) Error field primary edit rule
12) Error comments (supplemental information)
13) Item amount
14) Item ECE sequence number
15) Item routing number
16) Error internal identifier

Integration with Other Tools

Other tools also support your review and information export associated with errors:

- You can use the Excel Exporter to get a complete list of all errors and can then be saved or forwarded to others.

- You can use Print to get an error list in HTML format.
- You can use Export to get a list of all errors in CSV format which you can then import and process in other tools.

Cash Letter Summary

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**NO**	**NO**

The cash letters tab provides summary information for each cash letter that is present within the currently loaded file. Although most x9 files contain a single cash letter, there are implementations which use individual cash letters to separate and group items based on presentment requirements. Despite the reality that all x9 specifications support multiple cash letters within a single file, this facility is infrequently used.

A typical application is associated with x9 file aggregation. In that scenario, the current x9 file may contain information from a variety of other x9 files, where the original cash letters have been retained to explicitly define their origination and destination routing numbers. For those situations, this summary is very useful.

All columns in the cash letter summary are sortable, allowing you to quickly find any cash letter based on item count or amount. Error counters (error, warn, info) are provided which summarize the errors which were identified during validation. The summary includes the number of items and total amount. You can use the “line” column to resort into the original sequence. A “view” button is provided which allows you to drill down and launch to any cash letter within the x9 file.

Bundle (X9) column information is as follows:

- Cash letter identifier
- Record number
- Error severity icon
- Number of batches
- Number of validation errors, warnings, and informational messages
- Number of debits and credits (calculated from actual content)
- Total dollar amount of debits and credits (calculated from actual content)
- Destination routing and ECE institution routing
- Contact name and phone number
- Average front and back image sizes
- Maximum front and back image sizes
- Minimum front and back image sizes

Batch *And* Bundle Summary

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**NO**

The batch and bundle summary tab provides summary information for each batch (ACH) or bundle (X9) that is present within the currently loaded file. The batches tab is shown for ach files and includes specific columns based on that file format. Similarly the bundles tab is shown for x9 files and includes those columns that are appropriate when the file format is x9.37.

Error counters (error, warn, info) are provided which summarize the errors which were identified during validation. All columns are all sortable. You can use the “line” column to resort into the original sequence. A “view” button is provided which allows you to drill down and launch any specific bundle. Column information is as follows:

Bundle Summary (X9)	Batch Summary (ACH)
• Record Number • Cash letter identifier • Bundle identifier • Bundle sequence number • Error severity icon • Number of errors • Number of warnings • Number of infos • Trailer counter • Trailer image count • Trailer amount • Trailer MICR valid amount • Credit total • Debit total • Creation date • Business date • Destination routing • ECE routing • Return location routing	• Record number • Batch identifier • Service class • Entry class • Error severity icon • Number of errors • Number of warnings • Number of infos • Credit count • Credit total • Debit count • Debit total • Company name • Discretionary data • Descriptive date • Effective date • Settlement date • Addenda count • Entry hash • Company identification

Tiff Tag Summary

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**NO**	**NO**

The Tiff Tags tab provides summary information regarding all tiff tags that are present across all images within the current x9 file. The Tiff Tags tab is powerful because it provides a complete view of all tiff tags across all images. With this information, you can easily determine if tiff tags are being used consistently across all images within the file. Given how the information is summarized, it can also provide insight into possible errant tiff tags that might appear on a limited basis within a small number of images. This can be quickly determined based on the usage count that is present.

The following information is provided:

- Line
- TiffField (the tiff tag number)
- Description
- Usage (the number of times this tiff tag is present in the file)
- Errors (associated with this tiff tag)
- Severity (associated with any errors that were identified)
- Total bytes (the total bytes that are associated with this tiff tag across all images)
- Mandatory (indication if this tiff tag is mandatory per the tiff rules being applied)

If there are indicated errors, you can go to the Errors tab, find the error there (note that the Error tab can be sorted by clicking on column headings), and then launch to the error in question.

The tiff tags table is can be selected by both Print Tables and the Excel Exporter, making it available for export and sharing.

Filters

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Filters represent a subset of items within the currently loaded file. Any individual item can be only occur once within a filter, but can exist in separately defined filters at the same time. The items within a filter are arranged in a sequential order (relative to how they appear within the current file) regardless of the order that they may have been added.

The contents of each filter is shown as a separate tab within the dashboard. The filter list can be used to easily review all of the items that current exist within the filter, as well as to easily remove any individual items.

Filters can be created in multiple manners:

- Filter contents can be created manually using the picklist function from the toolbar. The item to be added would be selected within the file tree by clicking on them with the mouse. Multiple items can be selected at the same time by holding down the CTRL button and then selecting additional items. Once items are selected, the / filter / add to items picklist / function is used to add the selected to the picklist filter. Note that the picklist is operates like any other filter, it just has a unique (system defined) name.
- A filtered items list can be created using Find.
- A filtered items list can be created using Search.

Filters can then be maintained in variety of ways:

- Items can be removed (on at a time) from a filter list using the “remove” button.
- Entire filters can be individually deleted in their entirety.
- All filters can be deleted in a single operation.
- Multiple filters can be merged into a single new filter; any common (duplicated) items within the filter will be eliminated such that each item exists only once.

Filters can then be used for various purposes:

- Create File can be used to create a new file that contains only those items that exist within a selected filter (this uses the filter list for inclusion).
- Create File can be used to create a new file that does not contain those items that exist within a selected filter (this uses the filter list for exclusion).
- Delete items from the currently loaded file.
- Image Print the selected items.

Find

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

Find allows you to easily locate items (debits or credits) based on your criteria. This can be done across the entire file, or can be limited to one or more selected bundles. Find is designed to locate items. If you want to look for data in specific record types that are not associated with checks (or in any other record types), then you should instead consider use of the Search tool.

Find will allow you to select items based on any combination of the following fields:

- Amount or amount range
- Routing, account number and/or check serial number
- Sequence number or sequence number range

The routing number should be entered on a full nine (9) character basis. This includes the check digit field and should also include any embedded dashes. The search will be done using the routing number exactly as provided.

The check serial number will be used to search both the OnUs and the AuxOnUs fields. Items will be considered a match when your provided check serial number is located in either of those fields.

In addition to these very specific criteria, you can also select items at a higher level. For example,

- Debits only
- Credits only
- Only those items which contain errors

Using Find

Find remembers the last selection criteria that you entered within the current session and will retain that for your next find. You can reset back to defaults using the “refresh” button on the tool bar.

By default, Find will search across all bundles within the currently loaded file. You can use the bundles panel to instead limit your search to one more more specific bundles. After entering your selection criteria, you must then press the “Find” button to initiate your search.

The file viewer will position you on the first check that meets your specified criteria. You can then use the Find Next button (from the tool bar) to easily advance through the additional items. Note that the Find Next button will always indicate your relative position within the selected records. You are always informed of your current position and the total number of records that meet your defined criteria.

Creating Filters

You can build a filtered view using the “Filter” button. Filter uses the selected items to build and add a separate tab to the DashBoard. You can manually set the name that is assigned to the new tab, or alternatively allow Find to automatically assign the next logical tab number.

Creating filters provides several advantages and useful functions. Filter allows you to position on several records at the same time so you can bounce back and forth. An example would be having the primary tab focused on a bundle record while having the filter tab positioned on a specific check. Or you can position on different items within the two windows, which is helpful when you are looking for differences or duplicated items.

There are other advantages to using filters. For example, filters can be used by reporting and as the basis to print or save a list of the selected items based on your criteria.

Search

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Search is a powerful tool that can be used to search and optionally replace fields within the currently loaded file on a record by record and field by field basis. Search differs from Find in several ways:

- Search works at the record and field level, while Find is targeted for logical items.
- Search allows values to be replaced while Find does not.

Search String Entry

Search strings can be entered using one of several available options:

1) Actual value comparison (against each individual field value within each data record) using your entered search string. In this mode, there are two options that can be applied to assist in your search efforts:
    - Compare numeric fields on a logical value basis; enabling this option will remove leading zeros from numeric fields as part of the evaluation process (eg, a search string of “1” will match a data field that contains “00001”).
    - Compare alpha fields on an upper case basis; enabling this option will evaluate alpha fields using their upper case values, which would normally always be applied for data searches since the case of the data value is typically unknown.
2) Select all fields which contain blanks; these are typically conditional fields that are not populated with a value.
3) Select all fields regardless of value; this option must be used in conjunction the Field Selection panel. For example, this would allow you to select all Bundle Business Dates which you could then replace with an alternate value.
4) Compare on a contains value basis; a match is signaled when your entered string value appears anywhere within the string of characters for a given data field.
5) Compare on a leading value basis; a match is signaled when your entered string value appears as the leading string of characters for a given data field.
6) Compare on a trailing value basis; a match is signaled when your entered string value appears as the trailing string of characters for a given data field.
7) Compare on a RegEx basis (read on for a further explanation) where your search string is used as a RegEx expression against each data field; a match is signaled when evaluation of the RegEx expression results in a true value.

Replace String Entry

Replacement strings are entered using one of several methods:

1) On an actual value basis where you enter the actual replacement string to be assigned to each field. The new value can be purposefully blank, in which case you will be queried to confirm that you in fact do want to replace the current value with spaces.
2) On an actual value basis, where you enter the replacement string and an increment to be assigned to each field. For numeric or numeric blank fields, the increment is repetitively added to the values allowing them to be sequentially assigned. For string based fields, the increment is instead treated as a suffix which is appended to the end of the value string. In this situation, you can also control the minimum length of the suffix by entering leading zeros. For example, a value of "ID" with an increment of "0001" will assign string values of "ID0001", "ID0002", "ID0003", etc.
3) On a substring basis, where the replacement string is used to replace the first located occurrence of the search string within the current field value.
4) On a substring basis, where the replacement string is used to replace every located occurrence of the search string within the current field value.

Field Selection

Search allows you to identify the specific fields that are to be searched within the currently loaded file. The default is to search against all fields across all record types. You can use the field selection panel to limit the search to specific record types or specific fields using the provided check boxes.

To select all fields within a given record type, you must select the "xx.1" check box for that specific record type. For example, to select all fields within the type 1 record, you would select the "1.1" check box which will then automatically check all fields that exist for the check detail record type.

You can alternatively select on a field by field basis. This is a powerful option since it allows you to perform a very targeted search against one or more fields of your selection, by scrolling through the provided list and using the provided field level check boxes.

Search Actions

Available actions are as follows:

- ***Search***: initiate a new search using the current search string and options
- ***Next***: advance to the next hit with the current search list
- ***Replace***: replace the current hit with the replacement string and advance to next
- ***Replace All:*** replace all remaining hits with the replacement string
- ***Filter:*** creates a new filter that contains the selected items based on the current search

criteria

Filtering

Search allows an item filter to be created from the search results. Note that filters can contain items only, so any field level hits in header or trailer records will not be included in the created filter. Items are included in the filter when any field within the item group has been hit by the entered search criteria.

Use of the Modify Log

Search/Replace is integrated with the Modify Log, with each field replacement being logged there as a modified field just as though you used Modify to actually make the change. This design allows use of Modify (from the tool bar) to review all changes that are made using Search/Replace. Modify can then also be used to launch or even revert individual changes. The Modify Log can be printed to PDF or exported to Excel, providing a full audit trail of your changes.

Remember that files are actually stored within available memory and are not actually created (written) until you save to an external file. Hence you can always decide to NOT save your changes and hence any replacement actions will be lost. When saving files, you should always write to a new file name and not overwrite your original input file.

RegEx Search

What is RegEx? Per Wikipedia: “In computing, a regular expression is a specific pattern that provides concise and flexible means to "match" (specify and recognize) strings of text, such as particular characters, words, or patterns of characters”.

A full RegEx tutorial is beyond the scope of this document. However, you can use the Internet to search on “Java RegEx Tutorial” and you will find a lot of material that can assist you in the use of this powerful search tool. That said, the following is a list of commonly used search patterns:

Meta Characters	\d	Any digit, short for [0-9]
	\D	A non-digit, short for [^0-9]
	\s	Any whitespace character
	\S	Any non-whitespace character
	\w	A word character, short for [a-zA-Z_0-9]
	\W	A non-word character [^\w]
	\S+	Several non-whitespace characters
	\b	Matches a word boundary
Quantifiers	*	Occurs zero or more times, is short for {0,}
	+	Occurs one or more times, is short for {1,}

	?	Occurs no or one times, is short for {0,1}
	{X}	Occurs X number of times, {} describes the order of the preceding liberal
	{X,Y}	Occurs between X and Y times,
	*?	? after a qualifier makes it a "reluctant quantifier", it tries to find the smallest match.
Matching Symbols	XZ	Finds X directly followed by Z
	X|Z	Finds X or Z
	.	Matches any character
	[abc]	Set definition, can match the letter a or b or c
	[abc][vz]	Set definition, can match a or b or c followed by either v or z
	[^abc]	When a "^" appears as the first character inside [] when it negates the pattern. This can match any character except a or b or c
	[a-d1-7]	Ranges, letter between a and d and figures from 1 to 7, will not match d1
	^regex	regex must match at the beginning of the line
	regex$	regex must match at the end of the line

Excel Exporter

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

The Excel Exporter provides facilities to export virtually all reporting information tabs into Microsoft Excel or a compatible spreadsheet program such as Open Office. This is a very convenient way to export this information into a format which you can save for later reference, used for data analysis, or forwarded to others.

Excel Exporter presents you with a menu where you can indicate the tables you would like to select from those are available within the application. One or more filters can be selected. If you want to select more than one filter table, hold down the control (CTRL) key and use you mouse to select the desired tables.

Once you have selected your tables, press the format button to initiate the export process. The Excel Exporter will generate an XLS file within the /temp/reports folder with a uniquely assigned file name (which is accomplished by suffixing a date/time stamp to the file name that is created).

Once the output workbook is created, the Excel Exporter will then launch the created XLS file allowing you to view the results and take any additional desired actions.

There are program options that define how long generated reports will be maintained within the temp/reports folder. You can change this parameter with the default being three (3) days.

Excel Output

Each selected table will be written to a separate worksheet within the overall workbook that is created. You can view the worksheet names at the bottom on the Excel screen. These worksheets relate back to the various reporting tables that were initially selected. The results can be reviewed or saved in your own Excel output file and then subsequently saved to an external file name and format of your choice.

Print Tables

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Print Tables provides facilities to print the same reporting information that is created by X9Assist. Output reports are created in HTML format that will then be viewed using your default Internet browser. Reports are created and stored in your Documents/x9_assist/reports folder for viewing.

Print Tables presents a menu where you can indicate the tables you would like to select from those are available within the application. These are the same tables that can be viewed via the dash board. The X9Assist tables are shown in several columns for your selection:

- All reports that are associated with the currently loaded file.
- All reports that are associated with the record group that is currently positioned and being displayed within the record viewer. Including these tables are appropriate when there are issues with a specific record and you want to include that information in the reporting package that is created.

Printing From A Filter

Print Tables also allows you to select from one or more filters that you may have created for the current file. The contents of each filter will be printed just as they are displayed within the filter tab. This is an easy way to create a report containing specific items within the current file. One or more filters can be selected. If you want to select more than one filter table, hold down the control (CTRL) key and use you mouse to select the desired tables.

Creating the Output Report

Once you have selected your tables, press the format button which initiates the creation of the HTML based reporting file. When this is complete, X9Assist will launch your Internet browser to allow you to view the report file, which you can then browse, print, or take other actions such as email, save (etc). Reports can be printed to your desired printer as part of the viewing process.

Report Retention

X9Assist has options that define how long generated reports will be maintained within the temp/reports folder. You can change this based on your requirements, with the default being three (3) days.

Printing to a PDF

You can optionally print to an output PDF (and not to an actual printer) by installing one of many PDF virtual printer products that are available. One such example is “Microsoft Print to PDF”

which has been a Windows component since Windows XP; there are others. These PDF printers act as a logical printer and will intercept your generated image stream as your output is generated. By using such a tool, you can easily write your print stream to a PDF file. You can then take the PDF file into other applications and tools of your choice.

Summary

Print Tables is a powerful function that allows X9Assist created reports to be easily converted to HTML format which can then be browsed, printed, or emailed to others. The HTML output is generic in nature and can be used for many other purposes.

Print IRD

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**NO**	**NO**

Print IRD is available with our X9Validator and X9Assist products and provides a facility to attach the IRD creator routing and business date to image replacement documents that are to be printed. Input to Print IRD can be either the entire file or an input filter, where this selection is made through a drop-down box.

Filters can be created through a variety of functions:

- Find
- Search/Replace
- Picklist
- Returns

Print IRD can format items for one of two scenarios:

- Forward presentment replacement items
- Return items

If you need to print return IRDs for certain forward presentment items from an ICL, you should use our Returns function (on the toolbar) to build a filter of the items to be returned. This will allow you to enter the return reason for each item to be printed. In this way, the appropriate return reason can be applied to each item.

If you need to print return IRDs for certain return items from a returns ICLR, you should use the add to picklist function (on the toolbar) to build a filter of the items to be returned. This will allow the existing return reason for those items to be retained.

If you need to print all return items within an ICLR, you should select the “Current File” option within the drop-down box. This will allow the existing return reason for all items to be retained.

Once the items are formatted, processing will be transferred to Print Image, which is our common facility to print images. Please refer to that documentation for how Print Image is used.

Print Images

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**NO**	**NO**

Print Images is available with our X9Validator and X9Assist products and includes facilities to print check images in various page layouts which can be either system or user defined. A common use is to format and then print item images to a MICR printer in a three images per page layout. However, you can define and then print more complex scenarios of up to twelve checks per page in a user defined grid format. Print Images allows you to define and control the actual coordinates of printed images on the physical page, which can be either letter or legal and then either portrait or landscape.

Print Images requires that you first select the specific images to be printed:

- All all images that exist in this x9 file;
- Items defined within a filter which is created using tools such as find or search
- The current item displayed within the browser tree

Print Images then requires that you select an image print format from a drop down box which is populated with system defined format. Each format defines how many images will be printed per page and the spacing the images within each page. Several print formats are configured and included on a standard basis with X9Assist. You can also use the Print Images dialog to create and save your own print formats for subsequent future use. This is an important function since it gives you fully control your image print layouts.

Print formats define the attributes and page coordinates of your check images when they are transferred to the printed page. The print format contains various attributes that are used during the printing process (paper size, orientation, title, margins, and most importantly the actual x/y coordinates where checks will formatted and printed.

You can create new print formats by displaying an existing format, making changes to it, and then saving it under an alternate name. Print formats are stored in folder / Documents / x9_assist / xml/ printFormats /.

Predefined Print Formats

The following print formats are included in our distribution. These can be used as the basis to create new formats subject to your requirements. The base formats are as follows:

Format Name	Usage	DPI	Rows	Columns
Landscape 3 x 2	Print item images six per page.	600	3	2
Letter 2 x 1	Print item images two per page.	600	2	1

Format Name	Usage	DPI	Rows	Columns
Letter 3 x 1	Print item images three per page.	600	3	1
Letter 3 x 1 IRD	Print IRD images three per page where the MICR line will be redrawn from the x9 data; this format will print at 1200 DPI with options to maximize quality when printed items will be subsequently re-scanned.	1200	3	1
Letter 3 x 1 MICR	Print item images three per page where the MICR line will be redrawn from the x9 data; this format will print at 1200 DPI with options to maximize quality when printed items will be subsequently re-scanned.	1200	3	1
Letter 4 x 2	Print item images eight per page.	600	4	2
Troy 3 checks 8.5 x 11 DLA834 (7000)	Print IRD Images three page page using 8.5 x 11 form DLA834 (7000).	600	3	1
Troy 3 checks 8.5 x 13 45-20462	Print IRD Images three page page using 8.5 x 13 form 45-20462.	600	3	1
Troy 3 x 1 on generic 8.5 x 13 paper.	Print IRD Images three page page on generic Troy 8.5 x 13.	600	3	1

Print Functions

Once you have selected your items and print format, you can then initiate one of several available actions:

- ***Print***: open a print dialog and initiate printing to a selected printer.
- ***Zip***: write the formatted image pages to a user selected zip file.
- ***View***: launch an external viewer of the formatted image pages.
- ***Preview***: take a quick look at the formatted pages that are formatted for print.

View

View provides an interface to the system installed PDF or XPS printer and viewer. This is a two step process where the image pages are written to an intermediate file and then an external viewer is launched. In a Windows environment, the Microsoft Print to PDF write will be used to create an output file which will be launched in the default viewer on this system. This facility provides better

viewing than our preview facility, but does take longer given the extra time that will be needed to create the intermediate file. The functionality is excellent once the viewer is launched.

Preview

Preview allows you to take a quick look at the generated print pages. The next and previous buttons can be used to browse through the output. The page number box can be used to jump to specific pages. The zoom slider can be used to increase and decrease the zoom level of the current image, along with the reset button which moves back to a standard zoom setting.

MICR Line Redraw

Print Images includes enhanced functionality to redraw MICR lines (within each image) from the associated x9 item data (for example, the type 25 and 31 record types). This function should always be used for IRD print. It may also be applicable to items that are created by our Make/Generate functions, where the image MICR lines are known to match the individual x9 data records. MICR line redraw can be selected only when printing full scale images (when printed output has a single image column).

The purpose of MICR line redraw is to optimize the quality of the MICR lines when test documents are being printed for image enabled capture environments (Branch Capture, ATM Capture, Image Lockbox Capture, Remote Deposit Capture, etc). In those situations, it is critical that the MICR line be printed at the highest possible DPI supported by your printer, and not the 200 or 240 DPI that is present in an attached image per x9 image exchange standards. To meet this quality requirement, Print Images has the ability to directly redraw MICR lines from the x9 data. When doing so, the MICR line data from your images will not be used. This redraw function allows all MICR lines to be drawn at the current printer resolution (not from the image DPI) which can dramatically increase the quality of the scan line.

This functionality is appropriate only when performing IRD Print functions or when the printed images were created by the X9Assist Make or Generate functions.

MICR Line Redraw Usage Exceptions

There are certain conditions when MICR line redraw very explicitly cannot be used:

- The purpose of MICR line redraw is to improve the pixel quality of the E13B characters within the MICR line and to thus improve the MICR and OCR readability of those characters, with the intent to reduce digit misreads. There is no reason to use this functionality if the paper documents you are printing will not be subsequently recaptured.
- You can only use MICR line redraw when you are 100% certain that the x9 data matches the image data. This is typically NEVER TRUE for typical x937 files, since although the image data is typically very close to the x9 data, it is normally not an exact match. Hence this MICR line redraw does not apply to typical x937 files and should never be used in those situations. It can be used for files created by our Make/Generate functions, since that is a much more controlled environment where the x9 data and the image data should match.

- You similarly cannot use MICR line redraw when you are providing your own MICR lines into Make, which is typically done for negative use case testing of specific MICR line data content. In that situation, the x9 data will not match your MICR line data, hence the redraw function is not applicable.

MICR Routing Redraw in xxxx-xxxx Format

The option to redraw MICR routings in xxxx-xxxx format (from eight character routings) would normally always be enabled when MICR line redraw is active. This function will redraw a routing of "12345678" as "1234-5678", essentially inserting the dash that was removed during capture.

Some background may help here. This situation applies specifically to US routings, which would typically consist of nine digits which include a suffixed check digit and would never include an embedded dash. However, prior to the implementation of the MICR routing check digit, these numbers were instead formatted as "xxxx-xxxx" with an embedded dash, thus making up the same nine character string that exists today. Although most US routings now utilize the newer nine digit format, there are various situations where the older 4x4 format is used. This certainly includes old paper where customer accounts have not been updated to the nine digit format that includes the check digit.

The x937 file specification has the routing divided into two separate fields, which are the eight character routing followed by the check digit. When the routing was actually in 4x4 format (with the dash and without the check digit), then the x9 data is populated with the dash removed. In that situation, the eight character routing is present and the check digit is a blank to indicate that it was not present.

The bottom line is that these older routing numbers represent a special condition for MICR line redraw, since the x9 data does not actually contain the embedded dash. This option thus allows the dash to be reinserted into the MICR routing when it was removed during capture and is not present in the x9 data per the x937 specifications.

Printer Resolution

Print Image will automatically assign an appropriate DPI that is optimized based on the work being performed. Specifically, Print Image will enable 1200 DPI when printing items that are printed as MICR or IRD, to be subsequently scanned. subsequently processed electronically. The following print formats should be used:

Subject to the capabilities of your specific printer, all output is typically printed at 600 DPI. However, this should be increased to 1200 DPI when printing IRDs or items that were created by our Make function. This higher DPI setting is needed to minimize the potential for digit misreads of these printed documents. In these situations:

- The "Redraw MICR from x9 data" box should always be checked which will use the MICR line data from the appropriate record types (25, 31, 61, 62, etc) to reformat an improved E13B scan line.

- The “1200 DPI print is enabled as needed” box should also be checked, which will enable 1200 DPI printing.

Printer Drivers

In order to utilize our higher DPI print capabilities, it is important that you use a printer with an associated print driver that supports 1200 DPI printing. As an example, using an HP MICR Printer with the generic HP printer driver may well result in a configuration that does not support 1200 DPI printing. Although the printer is 1200 DPI capable, it may be that the printer driver is not.

If you use a printer driver that only supports 600 DPI, then the resulting check images will not be drawn at their expected size. This problem is a result of attempting to print at 1200 DPI with scaling that is ultimately not recognized by the printer. There are two ways to resolve this problem:

- On a short term basis, make sure you do not check the “Enable 1200 DPI” box, which will ensure that printing is at 600 DPI. Although this will resolve the problem, it is not optimal, since printed images will not be at the highest quality possible. This could possibly result in a higher digit misreads during subsequent scanning. However, note that it is commonplace for images to be printed at 600 DPI, so this should not create scanning issues.
- On a longer term basis, we recommend that you consider installation of a 1200 capable printer driver when you have a 1200 DPI printer. You can either use a specific driver provided by your printer manufacturer, or possibly use a generic driver (such as PCL or Postscript) that will work with your specific printer. You should review the technical documentation that from your printer vendor to determine your best options.

Additional important information:

- The Hewlett Packard “HP Universal Print PCL6” driver does not support 1200 DPI. Although this driver works well in a general office environment, it does not provide the higher resolution that is needed for MICR printing.
- The Richo “PCL6 Driver for Universal Print” driver includes support for 1200 DPI.
- The Xerox “Global Print Driver PCL6” driver includes support for 1200 DPI.

Print Formats

Image pages can be defined and saved for future reuse using various options and capabilities:

1) Identify which images should be inserted for each item (front or front/back)
2) Specify if you want descriptive captions printed below each item image
3) Select collated print format
4) Select two sided print format
5) Load a predefined format (these are saved in the /x9_assist/xml/printFormats folder)
6) Format in either a one column or two column grid
7) Format with up to six checks vertically
8) Define your image layout based on the number of rows and columns
9) Select paper size (letter or legal) and orientation (portrait or landscape)

10) Position the title, date/time, and page number based on your template
11) Modify the title for each print request
12) Save new or modified layouts to the /x9_assist / xml / printFormats folder so you can easily use them in future print sessions

Collating

You can optionally collate your images when they are printed. This is done to simplify the process of hand sorting the items after they are cut and printed. If you select collation, the image print sequence is modified so that you can cut the items by row and then easily restack them. This option will eliminate the more difficult process of handing sorting the items after they are printed.

Two Sided Printing

Checks can be optionally formatted for two sided printing. This option is typically not needed, since printers also support this functionality and will rearrange printed pages as needed based on their specific requirements. Our two side page sorting is provided only has a secondary function when absolutely needed. When two sided print is active, front images will be printed on odd pages and back images will be printed on even pages. The purpose of this page rearrangement is to allow checks to be printed and trimmed such that their front and back images are on the same piece of physical paper.

Cover Pages

Optional cover pages can be included in the printed output. These will be leading pages that include basic attributes of the input file (file name, folder name, file size, number of records, etc), along with key fields from the file header and trailer records. There will be one or more cover pages printed. The number of output pages is largely dependent on the number of batches that are included within the file.

Item Descriptors

Optional item descriptors can be included in the printed output. A descriptor box is formatted and inserted for each item, which includes information about the item itself (from the item record) and then from the attached addenda records. When item descriptors are enabled, other options such as two sided printing, collated, and image boxes should be turned off.

Creating Custom Formats

If you are creating a custom print format, you should follow these general steps:

- Select your paper size and orientation
- Use auto-layout to define the row / column configuration for the check images
- Indicate if you want front images only, or if you want to include back side images
- If needed, then modify your border sizes based on the PDF template

- Specify the location for your title, date/time, and page number
- Finally, adjust the locations of your check images based on your layout requirements

Image Size

Print Images will attempt to maintain the exact image size of your selected checks. Image size can be preserved based on the combination of image size, paper size, and orientation. For example, typical retail checks can be printed three per page using letter (8.5 x 11) with portrait orientation. Print Images will progressively reduce the resulting image size based on the parameters that you have selected.

Printing to a PDF

You can optionally print to an output PDF (and not to an actual printer) by installing one of many PDF virtual printer products that are available. One such example is “Microsoft Print to PDF” which has been a Windows component since Windows XP; there are others. These PDF printers act as a logical printer and will intercept your generated image stream as your output is generated. By using such a tool, you can easily write your print stream to a PDF file. You can then take the PDF file into other applications and tools of your choice.

Clipboard

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

Tables / Text

All of the tables that are generated by our desktop tools can be easily copied into the Clipboard. This facility can be very helpful when you would like to export information into another application for further analysis or sharing with others.

Here is an example of how the tiff tags for a specific image can be copied into the clipboard:

- First position on the item that you are interested in. Once positioned on the selected item, you would then launch the Item Viewer from the tool bar.

- Next, display the image tiff tags by toggling on the “Show Tiff Tags” check box which is in the action panel at the bottom. The tiff tags will now be displayed.

- To copy the tiff tags for the image, you when next select all lines in the Tiff Tag table. You do this by clicking within the table and using “CTRL A” on your keyboard. This will select all rows and all columns. You can then paste the selected rows from the clip board into your target application (such as Excel) by entering “CTRL C” from your keyboard.

- Finally, you can paste the selected text from the clip board into your target application (such as Word) by entering “CTRL C” from your keyboard.

This same copy and paste function is implemented in all tables through our desktop applications.

Images (X9 Only)

You can also copy images to the clip board but using a somewhat different procedure.

Again, you would first position on the item that you are interested in. Once positioned on the selected item, you would then launch the Item Viewer from the tool bar.

Then use the “Copy” button on the action panel at the bottom to actually copy the current image into the Windows clip board. You can then paste the selected image from the clip board into your target application (such as Word) by entering “CTRL C” from your keyboard.

Modify

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Modify provides facilities to make changes to individual records and fields within the currently loaded file. Changes are made to an "in memory" copy of the currently loaded file and will not be available to other applications until you save your results to a new output file.

It is highly recommended that you do not replace the current input file (by saving with the same file name), but that you instead create a new and unique file with your changes. This is a best practice that should be used whenever possible.

All file level validations will be run when you save a modified file to a new replacement file. You should always look at the results and attributes of the newly saved file very closely. This includes the number of records, dollar totals, and validation results.

Modify Sessions

Modify is an advanced facility which includes the ability to:

- Change individual fields within as many record types as desired.
- Delete records using several techniques (individually, by group, by record range, all good items, or all error items).

A modify session can be enabled or disabled at any time. Note that modifications to the currently loaded file are only allowed when a modify session is actually enabled. This is a preventative measure that requires that you first activate the modify facility before making changes.

You can easily tell when a modify session is active. The background color will change from the standard color (default is green) to an alternate color (default is turquoise). You can also visually tell which fields have been changed, since all modified fields will be highlighted using an alternate color as well (default is to show in a light orange). These colors can be changed through the program options panel based on your specific preferences.

Modify includes a counter in the lower left corner of the dashboard that indicates how many modifications have been made to the current file. This counter provides a quick and easy mechanism to visually determine the number of field changes that have been applied.

Modification Log

The modification log is a running list of all of the field level changes which have been made to the in-memory copy of the current file. The modification log includes information about each field which has been modified, including the before and after values for each such field. If a field has

been modified more than once, then the modification log will report the original value and the most recently assigned value for the modified field.

The modification log can be used to jump to any record that has been modified, allowing you to view the context of the change that has been applied.

The modification log will also allow you to revert any single modification. When a change is reverted, the original value will be applied and the modification will be removed from the log, just like it never happened.

Fields that can be Modified

The file specification (in xml) defines which fields can be modified. Typical defaults are assigned that allow virtually all fields within all record types to be modification. However, this facility can be customized by your organization should you want to limit the ability to apply certain modifications. This can be accomplished with an update to the various xml rule files.

Audit Trail

The modify panel will always provide a list of all modifications that have been made since your modification session was initiated.

Modify will track changes at the field level and will tell you if you have changed a field more than once during your modify session.

A modify report can be generated at any time using the reports facility (either PDF reports or the Excel Exporter).

The modification log can be written to an external file as a part of the save process. Please refer to save for more information about this facility.

Editing and Validation

When you enter a new value for a field, modify will apply all field level edits and will ensure that the specific value that you enter is acceptable. You will receive an error if the value is not appropriate for this field. Modify will justify the new value automatically based on the specification rules for the current field.

However, additional edits are required to cross check values against all other records. For example, if you change a truncation indicator, there are potential edits that will value this field against other record types (for example, an edit that would indicate that only one endorsement can have a truncation indicator of Y). You must rerun the validation process to have these edits applied. You will be reminded of this on the bottom of the DashBoard, where there will be count of how many changes you have made since the last validation was performed. You must run a validation prior to saving the file.

Modify Individual Record Fields

When you click the modify button for a given record, you are presented with a new panel that allows you to modify all fields within the current record within a single interaction. The modify record panel uses the defined alternate background color to highlight the fields that have been changed by earlier modifications applied during this user session.

This panel includes various information for each field that can be changed, including:

- Field number
- Field name
- Current value
- New value (this is the column where new values will be entered)
- Error message (when there are issues with the new value that you have entered)

You can use this panel to change one or more fields within the current record. You should pay very close attention to the “Error Message” column to ensure that the new values that you enter are acceptable per the edits that are identified for that particular field.

Note that you do not need to be concerned with justification, since positioning of the data that you enter will be automatically determined and applied by the modify process. You only need to enter the new value. Modify will apply all defined edits for the field being changed and will return any errors that are identified. You must correct errors before your new value will be accepted.

Overrides

There may be situations where you want to override the standard field modification process to allow you to enter values that would otherwise not be accepted. Overrides are provided in the upper left to allow you to activate these override functions, as follows:

- You can apply an override to allow invalid values to be entered. You would want to use this override when you are purposely creating invalid data for capture and downstream application testing. You can do this by selecting the override option which is at the top right of the modify record panel. By enabling this override, you can force a new value to be accepted by modify regardless of errors.
- You can apply an override to allow lower case values to be entered. Normally, all lower case values will be automatically translated to upper case. By enabling this override, you can enter lowercase values and they will be retained and assigned.
- You can apply an override to allow leading and/or trailing blanks to be retained as entered. The standard field modification process will remove all leading and/or trailing blanks that you enter and will then justify the field based on the validations rules that are defined for each field. By enabling this override, you are turning off standard data justification and blanks will be retained exactly as you have entered them.

Deleting Records and Record Groups

There are several delete strategies that are designed to simplify the process of removing records from the current file. These are as follows:

- Delete by record range. This technique can be used to easily delete a single record or a group of records (eg, an item, a bundle, or an entire cash letter).
- Delete all records by record type within range.
- Delete all records with a given record type.
- Delete all good items within the file thus leaving all error items.
- Delete all error items within the file thus leaving all good items.

When deleting by record number range, you should begin by positioning the viewer on the first record within the group that you want to delete. After that positioning is set, you then indicate that you want to begin the delete using the tools-delete function from the menu bar.

At that time, you will get the delete screen that is populated with information about the delete you just initiated. You should closely review this screen, including the record numbers, record types, and the number of records that have been chosen to be deleted. Based on your review, you can adjust the first and last record numbers within those respective fields.

You can use delete by range to accomplish many goals. For example, you can remove a single addendum record within an item. Or you can create an invalid file by removing a batch/bundle trailer while leaving the associated header and associated items. You can see the overall power of this function.

You obviously need to pay close attention to this record selection process. However, also be aware that you will have multiple levels of canceling your request. The most important is that you can review the change log (which will provide information on every delete you have done) and then will have to ultimately save the file for the changes to become effective.

Records can be deleted by record type. This is helpful in situations where you want to remove all records with the same record type (eg, you want to remove all addenda). This would be extremely difficult if you attempted to delete the records one at a time. You can alternatively use this facility to quickly delete all records by type in a single operation.

Finally, there is the ability to delete records based on their error status. You can also delete all error items (those marked with errors or warnings), or alternatively all items without errors (the good items). For example, this technique is commonly used to remove all good items from the current file and thus get a new file that contains only the error items so they can be more closely reviewed. In support of this capability, Delete also allows you to optionally remove all resulting empty bundles and/or cash letters from the file.

Delete Confirmation

Once you have reviewed your delete request and wish to proceed, you will next press the Delete button. When you do the delete, you will get a pop-up that must be approved for the delete to continue. Once deleted, you can then watch the modify log which will track all modifies and deletes that you perform during your session. You can also use the modify log to revert any individual changes that you have made.

Save

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Save is used to create a new output file from the currently loaded file. Save is most typically used after modifications have been applied. The save process allows you to specify the attributes of the new file to be created. This consists of setting the character set (EBCDIC or ASCII) and, for X9 Files, an indication of whether field zero lengths are to be included or excluded.

Save will inform you if the current file has validation errors, or if there have been modifications made to the file that have not yet been validated. These messages are informational. The recommendation is that you always correct errors before you save, and that you run Validate to be aware of the current validation status and that all desired errors have been corrected.

X9 files are normally written with standard attributes, which are EBCDIC and inclusion of field zero lengths. You should understand and consider the implications of not setting these attributes. Save will prompt you if you have decided to override these standards.

Once the desired file attributes have been selected, you then press Save and you will then select you the location and name of the output file that is being created.

Audit Trail

Save is fully integrated with the Modify process that tracks all modifications that have been made to the currently loaded file. Save informs you of the number of modifications that have been made, and allows you to optionally create and export the modification log as an audit trail for future reference. You can also optionally view the modification log after it has been created. Your options to both create the modification log and to view it will be recorded in user preferences, and then applied in future sessions.

Save has an optional property setting (within the X9Ware properties file) that allows you to specify the name of the modification log folder that is used on a mandatory basis within your organization. This is a configuration setting that is defined as part of your installation and cannot be altered or disabled. When the modification log property has been enabled within your environment, all modification logs are placed in that defined folder. Creation of the modification log will be done automatically for you as an audit trail. In this situation, you will not be prompted regarding the creation of the modification log. You still have the option to review the modification log after your save as been completed.

Output File Selection

Save will suggest a new file name that is created from the originally loaded file name. You can alternatively decide to overwrite the original (input) file, but that practice is NOT recommended

since it will destroy the original input file. When you do indicate that the input file should be overwritten, the save process will suggest that a backup be created of your original file as part of the save operation. Creation of this backup file is not mandatory but is highly recommended as part of your best practices.

Validation After Save

Validation will be automatically run against the saved file and it will then be loaded into the viewer. You should closely review the attributes and characteristics of the saved file. This includes the record count, dollar amount totals, and the validation status in terms of field level errors. These results should be logically compared against your original input file to ensure that all is as expected.

Repair

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Repair can be utilized to apply automated repair actions against the currently loaded file. Repair is designed to operate as a series of actions which can be applied to record types and their associated fields. You can select both the actions and the record types (as well as the individual fields) that are eligible for a repair operation. Repair actions have been purposely designed around situations where errors can be automatically addressed without negatively impacting the original intent of the file as created by the originator.

Repair is especially suited to correcting trailer records where the data is known to be correct but the totals in the trailer records are determined to be incorrect. In these scenarios, the trailer contents can be recomputed from the available data.

Repair is similarly well suited for “brute force” corrective actions such as correcting data alignment within fields, correcting truncation indicators, or blanking out user or reserved data fields when that is required.

Repair includes the ability to correct images. There are two available options:

- Replace errant images with missing image documents: Error images will be replaced with your defined “missing image” document. Note that this document can be customized as part of your product installation.
- Redraw errant images using the image data: Error images will be redrawn using the image data that is present, which will generate new tiff tags that meet required industry standards. If image inspection determines that an image is not usable, then it will instead replace that image with a “missing image” document.

Selective Record Type and Field Repair

Repair allows the scope of the repair actions to be limited to specific record types and/or fields. This is a powerful process since this allows the user to target all actions to a limited number of fields which will be targeted by the repair process.

Modify Integration

Repair is fully integrated with the Modify process. All field level changes that are created by Repair will create modification log entries. When Repair ends, you can view all of the field level changes that have been applied to your file by launching Modify and reviewing the log. This provides several advantages.

- You can both review and launch to any record that was modified by Repair.

- You can individually revert any change that has been applied by Repair.
- Finally, the modification log is available to your via Export or can be automatically created and written to a folder by Save.

Repair Results

On completion, Repair will provide a summary of the corrective actions that have been taken against the currently loaded file. This summary will include the total number of records that were repaired and then a list of repair actions by record type.

If record types are repaired that you have determined that you want excluded, you can revise your repair criteria and run the repair action again. To do that, you will need to reload the original file and start anew.

Repair will then launch a new validation process for the repaired file, which will provide an updated status on the errors that exist after the automated repairs were applied.

The repaired file remains in memory after the repair has been run. You must save the results to an external file if you determine that the results are beneficial.

Suggestions

We are interested in your suggestions regarding possible new repair actions that can be added to our automated repair process. Ideal repair actions must be generic in nature (applicable to all files) and must be associated with corrective actions that will result in the creation of a repaired file that meet industry standards and are acceptable to a large number of financial institutions.

Scrub

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Scrub is designed to remove varying levels of proprietary information from the currently loaded file by applying arbitrary and potentially random values to the customer confidential fields within the file. For X9 files, Scrub can also optionally perform replacement actions against the images associated with each item, allowing the image CAR/LAR to match the sanitized data content. Actions for a Scrub session to be save in a user defined scrub configuration file which is then available for future reuse.

The scrub process is very helpful when you want to use the currently loaded file as the basis to create a new output file to be used for testing or other internal purposes. This is a powerful facility, since the intent is to remove all possible customer data to create a new output file that has the same spirit as the currently loaded file, but with all proprietary information removed. Scrubbed files are very useful for testing environments where there is the need to minimize the use of actual customer data within the testing environment. Scrub is also very beneficial when you need to share the content of an existing file with a business partner, but you do NOT want to share customer information such as account numbers or check images.

Scrub has a standard list of available cleansing actions that can be selected based on your specific requirements. The current implementation is very robust however we realize that there is always room for improvement. Please let us know if you have additional scrub actions that you feel are missing that we can add to our standard scrub process.

Scrub will consistently update various records within the currently loaded file. For example, if you are scrubbing user or reserved fields, then those fields will be sanitized (replaced with spaces) across all record types that contain those specific field types.

Scrub Multi-File

Scrub within X9Assist runs on a single file basis as part of the user interface. However, note that scrub is also available from our X9Utilities product, where it can be run in multi-file mode, against a folder or group of folders. The scrub facility in X9Utilities is also multi-threaded for increased performance when running against a large number of files.

Scrub Actions

Scrub actions are used to direct the specific sanitization efforts to be applied:

- Scrub images
- Scrub item fields
- Scrub file headers/trailers

- Scrub amounts
- Scrub other headers/trailers
- Scrub addenda
- Scrub item sequence numbers
- Scrub date fields
- Scrub file ID modifiers
- Scrub user fields
- Scrub reserved fields

Field Inclusion and Exclusion

Records and fields are typically selected using the "Actions" related options on the left side panel. In some situations, certain fields must be added or removed from our standard scrub logic. This additional include/exclude functionality allows you to customize the scrub process by either adding or removing specific fields. Our design is to provide standard scrub support for all fields that contain personally identifiable information (PII) data. Because of that, this facility is most typically used to to exclude (remove) fields from the scrub that would have otherwise been selected.

The field inclusion and exclusion tabs are used to define exception fields which further tailor the scrub process to meet your specific requirements. These list define fields to be included (added) to the scrub process, or similarly fields to be excluded (removed) from the scrub process. These lists are saved with your scrub configuration and applied to future sessions.

By default, these lists will be disabled. This allows you to define a list which is saved with your scrub configuration, but then enable or disable the list when needed. The field lists are retained with your scrub configuration even when they are disabled. When you use this facilities, you must use the options panel (on the left) to active the particular list. Once active, these lists will be utilized by the scrub operation and then further identify targeted fields for the scrub process.

MICR Line Replacement

There are several options available for MICR line field replacement. When you replace MICR line data, you will typically always take some action against the images (since they contain the MICR data that is being sanitized). The MICR line replacement options are as follows:

MICR Replacement Option	Description
Replace both MICR OnUs and Aux OnUs fields	Numeric digits within the MICR OnUs and Aux OnUs fields will be replaced with newly assigned random digits. The format of these fields not be otherwise changed. For example, if the

MICR Replacement Option	Description
	MICR OnUs field contains a current value of "12345678/1234" it might be reassigned as "43861091/2863". The data length is not changed, only the digits that appear within the fields.
Replace the account number only	Numeric digits within the MICR account number will be replaced with newly assigned random digits. The number of digits in the account number remains unchanged. Other content with the MICR OnUs field is not changed. The MICR OnUs field must be properly formatted as "optional field 4 / account / process control" for this action to be performed.
Replace the entire MICR OnUs field	Numeric digits within the MICR OnUs field will be replaced with newly assigned random digits. The number of digits in the various OnUs fields remain unchanged. The MICR OnUs field must be properly formatted as "optional field 4 / account / process control" for this action to be performed.
Replace all MICR line fields from the use case file	MICR line fields are replaced using randomly selected entries from the provided use case file. The approach allows you to define MICR line data that properly applies to your test environment.

The use case file for Scrub contains the following information in this specific column order (with no column headings), that will be applied to check detail records. You can use the Use Case Creator (which is available on the tool bar) to create one or more such lists, and then select from those lists when you are running a scrub. You can alternatively create your own use case files using your own tools such as Excel. The use case file contains four (4) columns with data as follows:

- Column 1 – routing
- Column 2 – account number
- Column 3 – process control information
- Column 4 – auxOnUs

Scrub will randomly select accounts from your provided use case file and utilize that information to replace that corresponding data within the x9 check detail records. Each column within the use case file can be populated with a value, which is the replacement value, or can be blank, which implies that the original value should be retained. This approach allows account numbers to be changed while transaction codes (the process control field) is retained.

The use case file ensures that you have MICR lines that match your specific testing requirements (appropriate combinations of routing account, and check serial number). This ensures that the combination of routing and account is assigned, and that appropriate check numbers are used based on the account number (eg, if you need to test such things as controlled disbursements or drafts).

A use case can contain the Routing and Account Number only. In this situation, you can either leave Process Control and Aux OnUs empty. This format of the use case definition implies that you

do not want to alter these fields; their current value will not be changed (only the Routing and Account Number fields will be sanitized).

Images

Scrub has the ability to replace check images within the x9 file, which is critical when the MICR information and amounts on checks have been changed.

There are several options available:

- Draw images using the new MICR and amount information for each check: new images are drawn which will contain fields such as payee, date, courtesy amount, legal amount, memo, etc. The MICR line is obtained from the original image and will be drawn onto the new image within the clear band area. This allows you to eliminate a large amount of possible customer privacy information from the check since the redrawn image allows that data to be completely eliminated and replaced with a redrawn template image. X9Assist ensures that the newly drawn image meets tiff x9 exchange standards. You can use the image templates tab to select the business check formats (artwork) which will be used when dynamically drawing these images. X9Assist must use the business size checks to ensure that sufficient width exists to copy the MICR line data to the new image.

- Draw images while retaining the MICR line from the original images: images will be replaced with your defined “missing image” document. Note that this document can be customized as part of your X9Assist installation.

- Use missing image documents: images will be replaced with your defined “missing image” document. Note that this document can be customized as part of your X9Assist installation. The created missing image documents will include a MICR line that is drawn from the current x9 data.

- Scrub existing images using an XML definition that describes how input images should be manipulated to meet your specific business requirements: the image scrub process uses the original image to create a scrubbed image that retains portions of the original image. Parameters are then used to manipulate the image data to meet your specific business requirements while obfuscating as much data as possible. The image scrub process is applied to each incoming image (both front and back), allowing the physical image dimensions of those images to be retained. XML options are used to define the image manipulations that are applied, with separate actions defined for the front versus back images. Image scrub operates in one of two modes:

    (1) Begin with a blank image where all new fields will be drawn (including the MICR line) as a proxy for the current image. This is similar to our standard draw process but differs in to important ways. First is that the physical dimensions of each check image are retained. Second is that you cannot select the check design from a list of templates but must accept the standard template that will be drawn.

(2) Begin with the current image where specific boxes within that image data will be cleared. For example, this can be used to retain the entire front image of a check while clearing the customer name/address box in the upper left.

Scrub image XML definitions are located in folder / documents / x9_assist / xml / scrubImages /. When image scrub is selected, you must then also indicate which XML definition will be used to control the scrub process. There are a series of standard definitions that are provided by X9Ware which can be modified as needed. The list can also be extended with your own unique image scrub actions. Please see below for a list of the standard image scrub definitions that are distributed with X9Assist and the parameters that can used within these definitions.

Image scrub can retain the existing MICR line from the current check image, but be advised that this can be done only when the amount and MICR line data is not being modified by the current scrub operation. Retaining the MICR line has advantages when your exchange partners may not be consistently populating fields such as auxiliary OnUs in the type 25 records, which would prevent X9Assist from accurately drawing a new MICR line from the x9 data.

When retaining the MICR line, scrub will preserve the entire OCR clear band area at the bottom of each front image. This area is dedicated to the E13B MICR line on all check designs and should contain MICR data only. However, many checks impinge on that and may include lines and other check artwork within the clear band, even though they should not do so per defined standards. Customer signatures and other written notations may also appear within the clear band. Since scrub will retain the entire OCR clear band area, it is possible that some of this erroneous data is carried over into the new image that is drawn.

Scrub will copy the clear band to the new image exactly as it appears within the original image. On the SDK tab within program options, you can specify the size of the MICR line area that will be copied to the newly drawn check image, with the default to copy the bottom one inch. This preserves the MICR line and provides substantial information even if a strip has been applied to the bottom of the check as part of reject repair.

Scrub will inform you (as an error) when your scrub actions are not consistent with the image scrub definition that has been selected. For example, you cannot scrub MICR lines and then select an image scrub definition that retains the MICR line.

Scrub image ensures that the redrawn images meet tiff x9 exchange standards and that the front images created by image scrub will always match the associated x9 data.

Scrub Image Standard Definitions

The following standard scrub image definitions are distributed with X9Assist. Entries within this list can be modified per your requirements. You can also add to this list with new entries to meet your specific needs.

Scrub Image XML Definition	Front Image	Back Image
drawProxyRetainMicr.xml	Draw a proxy replacement image while retaining the original MICR line. The size of the retained MICR line defaults to 0.80 inch which can be overridden using program options. Text including CAR and LAR fields will be drawn using a generic handwriting font.	Retain the original back side image.
drawProxyDrawMicr.xml	Draw a proxy replacement image and draw the associated MICR line from the current x9 data. while retaining the original MICR line. Text including CAR and LAR fields will be drawn using a generic handwriting font.	Insert a drawn proxy as a replacement image.
drawMissing.xml	Insert a standard missing document as a replacement for the current image.	Insert a standard missing document as a replacement for the current image.
drawMissingRetainMicr.xml	Insert a standard missing document as a replacement for the current image while retaining the existing MICR line.	Insert a standard missing document as a replacement for the current image.
drawMissingDrawMicr.xml	Insert a standard missing document as a replacement for the current image while drawing the MICR line from the x9 data.	Insert a standard missing document as a replacement for the current image.
drawBlankRetainMicr.xml	Insert a blank image that retains the original MICR line. The remainder of the image (other than the MICR line area) will be blank. The size of the retained MICR line strip is assigned a default value to capture standard MICR lines and can be overridden using program options.	Insert a blank back side replacement image.
retainImageBlockImageAreas.xml	Retain the original image including the MICR line. Clear two areas within the image. First is the the name and address area located in the upper left hand corner of the check image. Second is a rectangle along the top edge of the check image.	Insert a blank back side replacement image.

Scrub Image XML

The following options can be specified in the scrub image XML definitions.

Xml Parameter	Usage	Values
retainSize	Retains the original image size and is applicable when either drawProxy or drawMissing has been selected.	True or false with a default of true.
retainImage	Retains the current image in its entirety. This parameter can be used in conjunction with one or more blockArea entries to clear possible confidential information within the image. The entire image is carried over if no blockArea definitions are present (which might be applicable to back side images).	True or false with a default of false. A value of false can be used to set the back side image to blank.
drawProxy	Draws a standard proxy image (including CAR and LAR fields) while retaining the MICR line from the current image.	True or false with a default of false. This parameter only applies to a front side image.
drawMissing	Draws a standard missing document. If this is a front image, then a MICR line can be optionally drawn or retained based on specified parameters.	True or false with a default of false.
drawBlank	Draws a blank image which will always retain the size of the original image. If this is a front image, then a MICR line can be optionally drawn or retained based on specified parameters.	True or false with a default of false.
retainMicr	Retains the original MICR line within an original image (which is the clear band area at the bottom of the image) but blocks the remainder of the image. This parameter is applicable only to retainImage and is needed since the position of the clear band varies based on check height.	True or false with a default of false. This parameter only applies to a front side image.
drawMicr	Draws a MICR line at the bottom of a proxy or missing image. The MICR line will be drawn in the clear band area using current x9 data.	True or false with a default of false. This parameter only applies to a front side image.
scrubBoxes	Defines a list of one or more boxes to be scrubbed. The contents of each of these boxes will be forced to white space.	Box list.

Xml Parameter	Usage	Values
scrubBox	Used when retainImage has been specified and is then used to define one or more boxes that are to be cleared with the check image. A blockArea parameter defines each box to be cleared. There are four parameters as follows: x coordinate of the box to be blocked specified in inches from the upper left hand corner; y coordinate of the box to be blocked specified in inches from the upper left hand corner; width of the area to be blocked in inches (defaults to the document width); height of the area to be blocked in inches (defaults to the document height).	Example as follows: <scrubBox> <x>0</x> <y>0</y> <width>3.0</width> <height>1.0</height> </scrubBox>

Scrub Image Exclusion by Routing

Images can be excluded from the scrub image process by routing number, which can include wild cards. This exclusion process is useful when you have internal routing numbers that identify internal documents that do not contain confidential information and hence do not need to be scrubbed. Typical examples of images that can be excluded would include GL tickets and Cash Tickets, which would not typically contain customer related information.

Images to be excluded from the scrub process are defined by routing in your scrub xml configuration. Each entry in this list should typically be nine characters and represents the routing as it appears in the x9 item record types (25, 31, 61, or 62).

The routing exclusion is variable in nature and can contain as many entries as are needed for your environment. Each entry in the list is matched against the item routing number. The dollar sign (“$”) can be used as a wild card character and is considered a match against any single character within the routing number.

Due to the variable nature of the routing exclusion list, it must be initially entered and then subsequently maintained using a text editor. Scrub user interface will recognize and inform you of the number of entries on the scrub panel. You should double check this count when you load your xml configuration.

A sample scrub image exclusion list is as follows:

```
<scrubFields>
        ….
        ….
</scrubFields>
<retailFormats />
<businessFormats>
```

```
  <checkFormat>business3.png</checkFormat>
 </businessFormats>
 <scrubImageExclusions>
  <routing>51???????</routing>
  <routing>52???????</routing>
 </scrubImageExclusions>
</x9Scrub>
```

Routing Replacement

Scrub will optionally replace bank routing numbers with new values at your direction. New routing numbers for individual items (debits and credts) are taken from the selected use case file. The routing numbers for other record types can be provided on the replacement tab. These values are saved to xml as part of your overall scrub configuration for future reuse:

- Origination ABAs
- Destination ABAs
- Return location ABAs
- Endorsement ABAs

Scrub Results

On completion, Scrub writes the sanitized data to an intermediate file and will provide a summary of the scrub actions that have been taken. This summary will include each selected action and the number of times that the action was applied to the current file. Scrub will then launch a new validation process against the scrubbed file, which will provide an updated status and allow you to browse the file and view the fields and images that have been substituted.

The scrubbed file remains in memory after the scrub actions have been applied. You must save the results to an external file if you determine that the results are beneficial.

Make

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

The task of creating both x9 and ach test files can become a very complex task. There are multiple issues and challenging problems that must be solved to allow the creation of these files. One must:

1) Define a tool to allow your test cases to be defined (either centrally or on a distributed basis), reviewed, corrected, maintained, reported against, and distributed as required.
2) Construct your items (debits and credits).
3) Create and populate addenda (endorsement) records to accompany your transactions.
4) Wrap the resulting transactions into bundles, a cash letter (for x9), and an overall file with the appropriate records, control totals, and file structure.
5) Build the data file with field values assigned based on length, justification, and pad characters into the target record types and formats subject to the rules specification.
6) For x9, also create industry conforming TIFF images that match the generated items so CAR/LAR can be run against the resulting file.

X9Assist is your tool to address these data generation requirements. The following tools are used to perform each of the above tasks:

Use case definition tool	MS-Excel
Build MICR transactions from your use cases	Make
Create addenda records	Generate
Wrap transactions and into a valid x9 file	Generate
Create conforming TIFF images	Import
Build the final file structure per specifications	Import

Make has been designed to simplify the creation of your test files. When paired with Generate, you can easily create test files from simple input files which can either XLS, XLSX, XLSM, or CSV. Make and Generate assume all of the tasks needed to format the data fields, construct the record types, and insert the needed headers and trailers with the appropriate control totals.

A Make run will always invoke our make, generate, and import processes. However, although these tools can be used individually, Make has been designed and implemented in a manner that allows your to view this as a single tool. This allows it to run in a single step, with the goal of reducing complexity and increasing productivity. Make can satisfy test document requirements for even the most complex testing needs based on the continuing enhancements that have been made.

For x9 files, Make/Generate can create output tiff images using either the E13B or CMC7 fonts to construct and draw the associated encoded MICR lines within the MICR band area.

Just to review how our various tools and how they are used:

- **Make** is used to create items (debits and credits) from your use case definitions.
- **Generate** is used to create various additional record types (addenda, image records, credit records, headers, and trailers) as needed to ultimately build a valid file structure.
- **Import (for X9)** is used to build the final output file by forming each data record per the requested file specification, assigning field attributes (lengths, justification, padding, etc) and ultimately writing the output file as both fixed and variable length records. This final step is not used for ach, since the file format is much simpler in nature.

You can view Make as a single tool that performs all of these functions. It is also interesting that each of these tools can be used independently for their respective functions. In this manner, X9Assist is designed to provide the appropriate tool to meet your needs.

Make/Generate as a Single Step Process

Although Generate can be run as a stand alone process, it is more typically run within a fully integrated Make/Generate function where they are run together as a single step. The Make panel allows you to first load your reformatter (which defines your Make parameters) and your generator (which defines your Generate parameters). When you run Make, you will see the Generate tabs are appended to the bottom of the Make tabs (on the right). The combined list of tabs provides all functionality to run Make and Generate within a single step operation.

Using Excel To Maintain your Use Cases

Why reinvent the wheel? MS-Excel is very commonly used within the industry. X9Assist takes advantages of your Excel knowledge by allow you to create, edit, and save your use case files directly as an Excel workbook. Make will read these workbooks directly without the need to convert them to some other format such as CSV. X9Assist supports both the XLS (Excel 97-2003) and XLSX and XLSM (Excel 2007) workbook formats.

Columns identifiers can be entered either as numeric values (eg, 1, 2, 3, 4…) or as Excel identifiers (eg, A, B, C, D ….). When you enter column identifiers in Excel format, they will be converted and stored as their corresponding numerical value.

As part of our Excel support, we also provide a VBA function that can be used to perform basic data validation as data is entered into your use case file. This VBA function has been incorporated into our ACH use case files, but is generic in nature and therefore could also be used for x9.37.

Using Excel Custom Number Formats

A number format is a special code to control how a value is displayed within Excel. Custom number formats can be used to control the display of numbers, dates, times, fractions, percentages, and other numeric values. Through the use of custom formats, you can do things like format dates to show month names only, format large numbers in millions or thousands, and display negative numbers in red.

You can optionally format your amount columns as numeric using decimals. If you provide a numeric field via a formula, it is important that you format that column as containing numeric data (and not general or text data). You can set the number of decimal places based on your formatting requirements (typically zero or two). If you use a format string against a numeric field, then Make will remove any commas, periods, or dollar signs that are present within the resulting string.

Generally, routing numbers are entered into your use case files as 9 digit numbers. However, there are times when it is desirable to explicitly format routings as either xxxx-xxxx (older US eight digit format) or xxxxx-xxx (Canadian format). X9Assist refers to these formats as 4x4 and 5x3, respectively. In these situations, it is recommended that these cells be formatted as text and not as numeric data with a custom formatting pattern. Using Excel text allows data to be entered on a WYSIWYG basis (what you see is what you get). By using this approach, you can define routing numbers in any desirable format as needed for negative use case testing (for example, 3x5 or 6x2). However, if there is a specific need to define routing numbers as numeric with a formatting pattern, then the following may be used:

Routing Format	Usage	Format 1	Format 2
4x4	US routing format	0000-0000	####-####
5x3	Canadian routing format	00000-000	#####-###

Other than the above stated exceptions, we do not recommend the use of custom number formats within your Excel use case definitions. Our alternative recommendation is that columns should be formatted as “Text” which means that the content will be treated as text even when a number is entered, and that the cell will be displayed and passed to Make exactly as it is entered. Using this best practice allows your fields to be processed on a WYSIWYG basis (what you see is what you get). Although Make supports the translation of numeric fields to strings through the use of Excel custom format strings, we do not recommend this only due to the complexities that are added to the process.

Using Excel Formulas

You can use formulas within your Excel worksheets to calculate values and feed those into Make. For example, you can use a formula to add up a list of items and provide that total into Make (also note that a deposit amount can be provided as zero, and Make will automatically calculate the tote for you). Another example of using a formula would be to inspect your ABA and transaction code combination, to assign the check format that you want to use for every item. Or you can define a formula to assign the memo field with your use case number. There are many potential uses formulas, allowing you to provide automation to your Make process.

Running Make

Make operates with several input components that must be selected:

- A reformatter which defines the rules on how to transform your use case file into the actual data that is needed by Generate. A high level description of this process is presented in the help section on the left hand side of the Make panel.
- A generator which defines the rules on how your items will be wrapped to ultimately create the desired output file. The selection of a generator within Make is optional. If you do not select a generator, then Make will interface to Generate, where you can subsequently define and invoke your generator. You will want to omit an generator the first several times that you use Make, to allow you to fully understand how Generate works and to build and save a generate configuration. Once you begin to get comfortable with the overall flow of the Make and Generate, you will be in a position to select one of your generators and then run Make as a single step (bypassing the Generate step from Make). You can only do this when you have defined and tested a generate configuration and fully understand the end to end Make/Generate process.
- A use case file which contains your specific routing/account use cases (these are your test cases). Each record (input row) of the use case file represents the data used to create a single check. You can build your use case definitions with our Use Case Creator tool, or you can build them from your own use cases, based on the actual data that you want to utilize for a given test. A use case file must be created in Excel format (XLS or CSV). Excel XLS files must be saved as an Excel 97-2003 workbook.
- You can also use the output files created by export as a use case file, where you would map the column definitions to those specific data elements.
- For x9, note that Make typically accepts the MICR OnUs field in its component parts (field4, account, serial, and transaction code). But it also allows you to specify a complete MICR OnUs field as well, which is how the type 25 and 31 records are formatted.

For each required check field that is manipulated by Make, you must specify one of the following:

- The use case file input column where this data will be taken. For example, if this is the ABA field and you have specified that in column one, then you will indicate that here. Note that column numbering is relative to one, so the first column is one (and not zero).
- The rules to be used to assign the data if it is not present in your use case file and it is instead to be statically or sequentially assigned. For example, if you use case file does not contain the item sequence number, you can specify a starting number and increment which will be used to create item sequence numbers and assign them to checks as they are created. For some fields on the Make panel, there is an option to generate all possible values, which means that Make will use the configuration rules to sequentially assign all of the defined values for this field.

Reformatters

Reformatters represent the power of the Make process since they translate your use cases into the format required by Generate. There are several reformatters that are included in the X9Assist distribution. You should review these reformatters since they represent strategies that can be used as a basis for developing your own use case reformatters.

Although you can write your own programs to create the native input required by Generate, Make can be used to virtually eliminate the need for that, by performing these data translations for you.

The following reformatters are included in the X9Assist distribution:

Type	Reformatter Name	Folder	Description
x9	default937.Reformatter.xml	/ x9_assist / xml / makeItems /	This is a basic x9 reformatter which is designed to work with a simple use case file that contains four columns: AuxOnUs, Routing, Account, and Serial Number. This basic reformatter is provided as an example but most probably will not be used in real world applications.
x9	depositReformatter.xml	/ x9_assist / xml / makeItems /	This is a more complex x9 reformatter which is designed to work with the depositFile.xlsx use case file, which is also provided with the distribution. All users tend to manipulate the reformatter content to meet their specific requirements, which is one of the benefits of using Excel as the basis for these documents. Once you understand how this reformatter works in conjunction with the sample use case file, you can then modify these examples for your organization.
ACH	There are a large number of reformatters for the ACH transaction environment, where they are created by standard entry class and file type (transactions, returns, dishonored, contested). These various reformatters are stored into sub-folders by their respective file types.	/ x9_assist / xml / makeAch / See the table below for a complete list of the standard ACH reformatters.	Similar to x9, these reformatters can be used as the basis to create your specific reformatters, based on your specific requirements. Because of that, you should not modify our standard reformatters. If you mistakenly modify a standard system reformatter, you can just delete it and X9Assist will automatically recreate it during the next user session.
CPA005	defaultCp5.reformatter.xml	/ x9_assist / xml / makeCpa005 /	This reformatter is used to work with Canadian electronic CPA005 files.

Saving and Reusing Reformatters

You can create new reformatters and save them for future use, and can assign your own name to the reformatter based on the purpose. This will allow you to build a library of reformatters that can be repetitively used for explicit test generation purposes.

Generators

Make/Generate is an integrated process where the reformatter and the generator are both accessible during the make process. Make allows a generator to be loaded, modified, saved, and ultimately used to create the output file. This process is extremely integrated.

Use of Heading Lines

Excel workbooks often include heading lines that provide instructions on usage. These are present as help (or tips) as to how the columns within the workbook should be populated. These leading lines within the workbook can be tagged with an “H” in the comment column to indicate that they are headings and not items. Excel conditional formatting can be used within your reformatter definition to color code heading rows based on presence of the “H” value.

Use of Comments

Comment lines are identified by placing an asterisk (*) in either the comment column or in the item type column, as identified by your reformatter. You can use the comment facility for various purposes. For example, a given item can be disabled (which will not be generated) by flagging it as a comment. Or, you can place comment lines within your use case file to describe the purpose of the test, information about specific test cases, or the originator of various bundles. Excel conditional formatting can be used within your reformatter definition to color code comment rows based on presence of the “*” value.

New Batches / Bundles

Items can be started in a new bundle in one of several ways:

- The item count per bundle is exceeded (as defined in your generator).
- The item is a credit, and your generator indicates that credits should begin in a new bundle.
- The item has a value of “B” in the “NewBundleOrCashLetter” column as identified by your reformatter.
- You populate differing values in Bundle Cycle Number, Bundle User Field, or both of these columns. A new bundle is started automatically when these columns are populated and the values for the current bundle have changed.
- You indicate that an item should begin in a new bundle using the item type indicator column, where you instead the keyword “Batch” (or “Bundle”) on a row by itself. When you do this, the row is treated as a separator line identifying the start of a new bundle, and will be otherwise ignored. The row will trigger the next item to begin in a new bundle.

New Cash Letters

Items can be started in a new cash letter in one of several ways:

- The item has a value of “CL” in the “NewBundleOrCashLetter” column as identified by your reformatter.
- You indicate that an item should begin in a new cash letter using the item type indicator column, where you insert the keyword “CashLetter” on a row by itself. When you do this, the row is treated as a separator line identifying the start of a new cash letter, and will be otherwise ignored. The row will trigger the next item to begin in a new cash letter.

Character Set and Print Fonts

Make uses the ISO 8859-1 character set to internally represent strings that can be drawn within item images. ISO 8859-1 is an 8-bit, single-byte coded graphic character set that encodes what it refers to as "Latin alphabet no. 1", consisting of 191 characters from the Latin script. ISO 8859-1 is commonly used throughout the Americas and Western Europe and specifically includes the French diacritical accents.

Any character that is included in the ISO 8859-1 character set can be utilized within your use case files and can be included within images drawn by Make. This is especially useful when creating data string that will be used for the Payee Name field of a Make item. You can search for topic “ISO/IEC 8859-1” on Wikipedia to get more information on ISO 8859-1 and the specific characters that are included in this character set.

Also remember that for special characters to be drawn within your images, they need to first be included within ISO 8859-1, but then also defined in the font that is defined for that particular field within your Make template. These templates are system defined but can also be modified to meet your specific needs. Templates are located in folder / Users / YourUserId / Documents / x9_assist / xml / templates /. There are separate folders by category (eg, retail, business, etc). Here is an example of the Payee Name field on one of our templates, showing that it is printed using font “handwriting5.ttf”. This font is internally defined within the X9Assist program launch folder. You can modify templates to use other fonts subject to your specific needs.

```
<field>
<fieldName>payeeName</fieldName>
<x>0.90</x> <y>1.12</y> <length>3.00</length>
<font>handwriting5.ttf</font> <fontSize>60</fontSize>
</field>
```

To test the ability to print specific characters, a good approach is to start your testing using an application such as MS-Word or Libre Office-Writer. Create a new document and enter text that contains the specific characters that you want to print. Select this text and change it to the font that you want to use for this Make field, that is defined within your Make template. You can now print this test page to determine if the text will print as expected. If any characters are translated to “?” then they are not supported by the chosen font.

Use Case Files

Our general intention is that you should not modify our templates, but alternatively use them as the basis to either save or further develop your own spreadsheets subject to your specific requirements.

- There are two X9 use case file examples. One is very basic (useCase100) while the second is a more “real world” example (depositFile.xlsx).
- There are also several ACH use case examples, which represent all commonly used NACHA transaction formats. These templates can be used as provided, or can also serve as a basis for developing your own use case templates. Where appropriate, these use case files will create balanced files, by automatically inserting an offsetting item which will net the file to zero. These templates have separate tabs (worksheets) for various standard entry classes. The reason for this is that the columns within the templates vary based on the specific data requirements. Because of this, you cannot combine multiple standard entry classes into a single worksheet. If you need to create an ACH file with more than one standard entry class, then the files should be created separately and then merged.

Our standard use case templates (included in the X9Assist distribution) are stored in folder / x9_assist / lists / useCases / as follows:

File Type	Use Case Name	Description
X9	useCase100.csv	A very simple use case file that is designed to work with default937.Reformatter.
X9	depositFile.xlsx	A more complex use case file that demonstrates some of the more advanced capabilities of Make. This example shows the power of using Excel and should be used as the basis for any use case files that you create.
X9	outOfFileReturns.xlsm	A specialized use case file that allows IRDs to be printed strictly from data entry and images. This reformatter can be used to print IRDs when you have the front/back images in your data archive, but you do not have an x9.37 ICL/ICLR file that contains the items.
ACH	ach_01_ACK.xlsm	Sample transactions for these entry classes.
ACH	ach_02_ARC_BOC_RCK.xlsm	Sample transactions for these entry classes.
ACH	ach_03_ADV.xlsm	Sample transactions for these entry classes.
ACH	ach_04_ATX.xlsm	Sample transactions for these entry classes.
ACH	ach_05_CCD_CIE_DNE_PPD_WEB.xlsm	Sample transactions for these entry classes.

File Type	Use Case Name	Description
ACH	ach_06_COR.xlsm	Sample transactions for these entry classes.
ACH	ach_07_CTX_ENR_TRX.xlsm	Sample transactions for these entry classes.
ACH	ach_08_MTE_POS.xlsm	Sample transactions for these entry classes.
ACH	ach_09_POP.xlsm	Sample transactions for these entry classes.
ACH	ach_10_SHR.xlsm	Sample transactions for these entry classes.
ACH	ach_11_TEL.xlsm	Sample transactions for these entry classes.
ACH	ach_12_TRC_XCK.xlsm	Sample transactions for these entry classes.
ACH	ach_13_IAT.xlsm	Sample transactions for these entry classes.
CPA005	cpa005.xlsx	Contains sample CPA005 transactions.

The ACH templates have multiple tabs for transactions, returns, dishonored, and contested-dishonored. You need to use the appropriate tab that matches the reformatter being used. For example, for a given entry class, the returns use cases with the returns reformatter. You can select the data type worksheet when a given XLSM workbook is opened.

Printing Out Of File Item IRDs

Make includes a predefined process that can be utilized to print IRDs when you have the front/back images in your archive, but you do not have an ICL or ICLR file that contains those returns. This process can be used to print forward presentment original items as forward replacement original IRDs, or return items as subsequent IRDs. The process consists of the following steps:

1) Look in your folder: / Documents / x9_assist / lists / useCases/, where you will need to have file outOfFileItems.xlsm. If you do not have that file, then you need to upgrade to the most current version of X9Assist, which can be downloaded from our website.
2) Look in your folder: / Documents / x9_assist / xml / makeItems /, where you will need files irdOutOfFileOriginals.reformatter.xml and irdOutOfFileReturns.reformatter.xml in your makeItems folder. If you do not have those files, then you need to upgrade to the most current version of X9Assist, which can be downloaded from our website.
3) Look in your folder: / Documents / x9_assist / xml / generate /, where you will need files irdOutOfFileOriginals.Generator.xml and irdOutOfFileReturns.Generator.xml. If you do not have those files, then you need to upgrade to the most current version of X9Assist, which can be downloaded from our website.
4) Using outOfFileItems.xlsm as a basis, create a new file of the items to be generated by this overall process.
5) This file should be similarly saved into the useCases folder. Please do not change our original template file. Instead, you should always make copies of this and adjust as needed for your organization. We highly suggest creating a new Excel file for each new IRD print run.

6) Use the Validate button to verify that the entered data is formatted correctly. The validate button can be used repeatedly as data is entered, just to confirm that there are no errors.
7) Save when your items are entered and validated.
8) Start X9Assist.
9) From the toolbar, run Make (which is the lightning bolt) on the toolbar.
10) Load the reformatter, which is either irdOutOfFileOriginals.reformatter.xml or irdOutOfFileReturns.reformatter.xml.
11) Load the generator, which is either irdOutOfFileOriginals.Generator.xml or irdOutOfFileReturns.Generator.xml.
12) Load the use case file, which is will be your items file as entered via Excel.
13) Verify that there are no input data errors.
14) Press the make button at the bottom of the Make panel.
15) Review the file that is created. This will be an ICL (originals) or ICLR (returns).
16) From the toolbar menus, run / Print / Print IRDs /.
17) Review options and then press the Format button.
18) You are now positioned within the Print Images panel with the “PrintList” selected (this list was automatically created by PrintIRDs.
19) All of the options are pre-selected for IRD print in 3x1 format on 8.5x11 paper; other print layouts can be selected here and even created as needed.
20) Preview (if you want) to take a look at the formatted IRDs.
21) Select Print which initiate the printing process.
22) Once completed, you most probably want to save the xlsm use case file to a unique file (using a time stamped name, or whatever) so you can reference and reprint these in the future.

ACH and Reformatters Use Case Templates

As you can see above, each use case file template can be utilized to create transactions for one or more standard entry classes. The standard entry class will be assigned by the chosen reformatter, and not within the use case file.

If you decide to create your own customized reformatters, you can utilize our standard reformatter for that entry class as the basis and starting point. There may be fields that will always contain a constant value, and if so, they can be moved into the reformatter and eliminated as a column within the use case file.

ACH entry classes are assigned to predefined templates as documented in the table below. For example CCD is assign to template T05 for transactions, R05 for returns, D05 for dishonored returns, and C05 for contested dishonored returns. Within a given XLS use case file, the template identifier must be present in cell B1. This setting identifies the template column definition, and is used by Make to ensure that the reformatter matches the use case file that is being read.

Entry Class	Transactions	Returns	Dishonored	Contested
ACK	T01			
ADV	T02			
ATX	T03			

Entry Class	Transactions	Returns	Dishonored	Contested
ARC	T04	R04	D04	C04
BOC	T04	R04	D04	C04
RCK	T04	R04	D04	C04
CCD	T05	R05	D05	C05
CIE	T05	R05	D05	C05
DNE	T05	R05	D05	C05
PPD	T05	R05	D05	C05
WEB	T05	R05	D05	C05
CTX	T06	R06	D06	C06
ENR	T06	R06	D06	C06
TRX	T06	R06	D06	C06
COR	T07			
MTE	T08	R08	D08	C08
POS	T08	R08	D08	C08
POP	T09	R09	D09	C09
SHR	T10	R10	D10	C10
TEL	T11	R11	D11	C11
TRC	T12	R12	D12	C12
XCK	T12	R12	D12	C12
IAT	T13	R13	D13	C13

ACH VBA Script

Our standard ACH use case files include a VBA script that can be invoked using the “Validate” button. Typically, data validation will be done (using the validate button) as part of the save process, before you exit Excel. Transactions must be located in the “Data” tab. If errors are found during the validation process, they will be loaded into the “(Errors)” tab. The validation script uses the “FIELDS” row as a definition of the attributes of each column within the worksheet. Data validation begins at the first row after the FIELDS row and continues through the remainder of the sheet. Commend rows (with * in column one) and heading rows (with H in column one) are excluded from the validation process.

The FIELDS row defines the attributes of the data that can be entered into the corresponding column. The value for a given column will consist of a number followed by zero or more data attribute characters. The number represents the maximum number of characters that can be entered into this column. The attributes can be:

- N = numeric
- R = required
- A = amount
- D = date formatted as yymmdd
- Y = date formatted as yyyymmdd
- M = date formatted as mmdd
- J = date formatted as ddd (julian)
- T = time formatted as hhmmss

Examples of column attribute strings are as follows:

- 10 = maximum of 10 characters
- 10R = maximum of 10 characters and the value cannot be blank/omitted
- 10NR = maximum of 10 characters and the value must be numeric
- 10A = maximum of 10 digits as an amount, which may contain a decimal point
- 10AR = maximum of of 10 digits as an amount, and a value must be entered

Creating Balanced ACH Files

ACH files can be either “balanced” or “unbalanced”. Balanced files contain an offsetting entry that automatically credits/debits the customer’s offset account for the debit and/or credit transactions that are present within the file. A debit file will contain an offsetting credit, while a credit file will contain an offsetting debit. From the bank’s perspective, a balanced file nets to zero, since the total amount of the transactions are offset by the balancing entry, which means that the net of debits and credits will be zero. Unbalanced files do not have this offsetting entry. Unbalanced files can be all credits, all debits or a mixture of debits and credits that probably do not balance against each other. When an unbalanced file is generated,, the receiving financial institution must make a manual entry to credit/debit the customer’s account for the activity. The bottom line is that transaction files must ultimately always net to a logical zero. The only question is where the balancing entry will be created. Typically, if your financial institution has required that you prefund ACH Activity, you will be required to submit unbalanced. Most financial institutions require balanced files since the files are self-posting, manual entry mistakes are eliminated.

Our use case files are created using Excel, which makes the creation of a balancing entry relatively easy since an Excel formula can be used to automatically calculate the offsetting transaction that is required. There are several important considerations for this process:

- The balancing entry should typically be the last item within your transaction file.
- The balancing entry is defined by putting the string SETL in column one.
- The balancing entry will be created in its own batch, making it readily identifiable.
- The balancing entry amount can be assigned explicitly or via an Excel formula.
- The balancing entry will have a system created description of “Settlement Entry”.

Use Case File Tips

The following tips are helpful when creating your use case files:

- The comment column must always be the first column (“A”) within your document.
- Header lines (those at the top of your spreadsheet) can be identified with an “H” in the comment column. This is done by tradition as a unique way to tag those lines.
- Comment lines (any row that is not considered to contain use case data) should have an asterisk (“*”) in the comment column. For example, you can use comment lines to separate your data into logical groups or further identify the source or purpose of specific test cases.

- A spreadsheet row that has absolutely no populated data will be discarded and will not create a resulting item.
- Identify each individual transaction row as either a debit or credit (D=Debit, C=Credit). A row will default to a debit when no value is present.
- Note that you can have credits defined throughout your Make file, and you can also have multiple credits offset by your debits when that is a requirement. You can complete control over the order of the individual items.
- Items will automatically fill each batch/bundle unless overridden in one of several alternative manners.
- Within Generate, you can indicate that each credit should begin in a new batch.
- You can also insert a row within the use case file that has “batch” in the item type column, which will forcibly begin a new batch at that point within the generated file. In this scenario, the use case row is blank except for the inserted bundle identifier.
- Define all needed item related fields and assign appropriate values.
- For x9, define the Payee, Memo, and name/address text strings that will be inserted into virtual images as they are dynamically drawn.
- For x9, define the specific check format that you want used (if you leave this column blank, then the check format will be sequentially assigned by Generate).
- Consider using customized Excel calculations to automatically assign values based on your specific requirements; for example, you could assign the transaction code automatically from the ABA column.

Specific requirements regarding the creation of your use case files:

- You can provide a fully defined MICR OnUs field or the individual field components. If you provide the fully defined MICR OnUs, that value is taken first and will override any values that you provide for the individual fields.
- If you provide both Aux OnUs and a check serial number, then the Aux OnUs field is taken first and your serial number will not be used.
- If you need an eight digit ABA value encoded, then you need to provide that value with the dash in the ABA column. For example you must provide the value of 1234-5678. Providing the actual dash in the ABA field in the indicator that you do not have an ABA check digit.
- Make has extensive support for the type 68 user record. Please refer to that topic for specific information on how to define and create your custom user records.
- Make includes support for the fixed and variable formats of the type 27 and type 34 addendum records. Please refer to that topic for more information on creating those record types.

Process Control Field (For X9)

Most banks use the process control field for two separate purposes. The first is to specify an optional transaction code that for their internal processing. The second is to provide the check serial number on consumer (retail) checks.

- The transaction code is typically one or two digits.
- The check serial number is typically four digits.

- Your reformatter can define creation rules for both your check serial number and the transaction code. Note that both of these are optional, so either one or both may be present.
- The process control field can then be constructed as a combination of your check serial number and your transaction code. An option is provided to allow the transaction code to be appended to the left or to the right of your check serial number.

OnUs Field (For X9)

Make allows you to construct the OnUs field by defining several fields which will then be combined (optional field 4, account number, serial, and transaction code). As an alternative, Make allows you to provide the OnUs field in its entirety, formatted exactly as you want it. This does require that you provide the separator symbols ("/") between the various logical elements within OnUs.

MICR Line (For X9)

Make normally constructs the MICR line from the individual use case fields (AuxOnUs, EPC, Routing, OnUs, and Amount) that are then formatted per standard MICR line construction rules. There may be times when you want to assume control of MICR line formatting. This type of override processing is typically used in support of negative testing where you need to explicitly format the MICR line. Several examples are:

- Your application requires non-standard MICR lines.
- You want to purposefully create images (and perhaps printed items) with invalid MICR lines for negative use case testing.

In support of this, Make allows you to format your own MICR line which can be provided via your use case file. User input is typically limited to valid E13B characters. Make does not block the presence of other characters which may be useful for specific negative testing. The following E13B characters would typically be present in your use case file:

- 0 through 9 (the numeric MICR line digits)
- A (transit symbol)
- B (amount symbol)
- C (OnUs symbol)
- D (dash symbol)
- b (a single blank)
- * (digit error)

An example of a user provided MICR line would be as follows:

A057770930Ab20915353C7837bbbbbbB0000010000B

CMC-7 MICR Line Encoding (For X9)

Make/Generate has the ability to create test items using CMC-7 item level encoding, which is based on the CMC-7 encoding standards as documented by ISO 1004. The CMC-7 font is widely used for checks and drafts throughout Europe, South America, and various other countries around the world. The CMC-7 character set consists of the ten numbers (0-9), five special symbols (S1 through S5), and twenty six letters.

Each country has their own unique layout for their encoded CMC-7 MICR line which incorporate their unique data elements and encoding. These layouts vary due to the internal identification and routing requirements as defined by each respective country. Because of these variations, it is important that you understand the CMC-7 requirements for the countries to be generated.

Make/Generate accommodate these differences through two facilities:

- Make allows you to take complete control of the MICR line format at the item level. For CMC-7, this includes use of the S1 through S5 special symbols (internally defined as A through E) within user provided MICR line data.
- Generate allows you to select the MICR font to be used for MICR font to be used for encoding. This font selection is made on the “image” tab.

Examples of CMC7 formatted MICR data lines which would be provided as input to Make and then used by Generate to format items are as follows (note the use of A-E and b for blanks):

C4440101bC123456789123Eb111122223333A
C4440102E0182Cb4014Eb111122223333Bb4227A

Credits (For X9)

An input file to Make/Generate typically has a single credit (which offsets the checks that are present), but in more complex testing scenarios you may have multiple credits present. There are two approaches that can be used to create credits within the Make process:

- Include the credit(s) in your Make/Generate input. When using this approach, you have total control over the order of the credits relative to the debits, and you also have total control of the amount that this present on the credits themselves. Remember that credits are identified within Make use case input using the "ItemType" column, which must be populated with "C" to indicate that an individual item is a credit (and not a debit which is a "D"). To take absolute control over the credit amount, you can explicitly include the required amount within the credit item. In this situation, your provided credit amount will be used (it will not be calculated). You can alternatively set the credit amount to zero, which indicates that you want the amount to be auto-calculated by Make/Generate, based on the offsetting debits which are present. Use of auto-calculate is required if you are having Make/Generate insert the credit into your ICL file (since the credit does not otherwise appear). It is also helpful even when you include the credit in your use case file, when the checks may change from test to test and you always want the file to be in balance.

- Your generator can indicate that an offsetting credit should be automatically created and inserted. When this is needed, indicators must be set on two separate Generator panels. First is on the Generate tab itself, where you must indicate that credit (type 61/62) records are to be created. Second is on the Credits tab, where you must indicate the attributes of the credit record to be manufactured and inserted. When using this approach, there will be a single credit which is inserted using the defined attributes to offset the debits which are present in your use case file. The inserted credit will be immediately after the bundle header (this sequence is typically described as credits first). The credit amount will be automatically calculated from the offsetting checks. This is typically defined as an ICL (image cash letter) which contains a single credit offset by the presents checks.

Generate also supports multiple credit scenarios. When this is needed, the credits must be present within your use case file. They cannot be automatically created and inserted by Make/Generate, since their exactly position would be unknown. In this situation you can have a series of transactions, each of which consists of a credit offset by its corresponding debits. Depending on your requirements, there can also be more complex transactions, where you have multiple credits offset by multiple debits. Obviously, in these complex scenarios, you use case input is responsible for both the credit amounts as well as their logical sequence.

Depending on your requirements, credits may or may not require images. If images are to be created, you must indicate this on the Generate tab using the “create credit images” check box.

Addenda (For X9)

Make/Generate supports the creation of both primary (BOFD or Bank of First Deposit) and secondary endorsements, which are attached to each generated use case item. It is important to remember that creation of endorsement records must be enabled on the Generate tab when using these facilities. No addendum records will be created if those corresponding check boxes have not been enabled.

The primary endorsements are record type 26 (for ICL) and record type 32 (for ICLR). The secondary endorsements are record type 28 (for ICL) and record type 35 (for ICLR). Record types 26/28 will be attached to the type 25 check detail record. Record types 32/35 will be attached to the type 31 return detail record.

The most typical use of Make/Generate is to manufacture specific test data that evolves around use case items. Within this underlying purpose, endorsements are created as necessary attachments since they will be required by those image enabled applications which process the generated files. These creation of these endorsements is associated with the need to create a logically complete file, and less about the specific content of the endorsements themselves. To that end, Make/Generate provides facilities to build both the primary and secondary endorsements around these specific needs.

Items can optionally have a single primary endorsement using a specified routing. All primary endorsements will have this same routing. By default, the first endorsement will always have the

truncation indicator set to “Y”. This default behavior is can be disabled using a check box on the addenda panel.

Items can optionally have multiple secondary endorsements (from 0 to 99). The routing can be either assigned from a specifically assigned number, or can be randomly selected from a user provided list. Generating a single secondary endorsement with a fixed routing is the most commonly used approach for testing. However, in more complex situations, the routing can be assigned from a user constructed list which represents your exchange partners. Assignment from the list allows files to be generated the mimic your production environment.

Invalid Image Test Cases (For X9)

Generate has many capabilities, one of which is to create invalid image test cases. Our invalid image function provides a mechanism to create flawed images that can be sent to your capture systems to see how they react to these various scenarios. The list of invalid image variants has increased over time and we are open to suggestions as to improvements for this facility.

To use this function, you must select the “draw new images” option on the parameters panel (on the left). This process has numerous available options:

- You can select the invalid image test cases to be generated.
- Check boxes are provided which allow you to easily select all tests or entire groups.
- You must indicate the number of images to be generated per test case, which will default to one when not entered.
- The invalid image scenario is incorporated into the front side image. Depending on the test case, the resulting image may or may not be be viewable. This is dependent on the type of structural error which has been selected for that image.
- The invalid image test case name is drawn into the back side image which is otherwise not modified by this process. This means that each invalid image test case can be easily identified by looking that the corresponding back side image.

Using Make to Create Very Large Files

Make can be used to create very large x9 files. Although Make does not have an internal limitation on the number of items within a single file (within reason). While the XLS file format is limited to a maximum of 65,536 rows, XLSX and CSV do not have this limitation. Thus, those data formats can be be used to represent use case definitions with a very large number of items.

View

Once you define your reformatter rules, you can use the “View” button to view the checks that would be created by Make. This preview allows you to check your work before you actually generate a file. The test results are displayed within the Text Editor in CSV format.

You should always preview the output from your new reformatter rules to see if they are creating checks in the manner that you expect, prior to actually running generate.

View will display the Make created fields as they have been manufactured per your defined parameters. This field list represents the internal interface from Make to Generate.

View allows you to see your Make results without actually running Generate. This allows you to view your field mapping to ensure that you are creating your fields as expected. You can make adjustments to your mapping and repeat this process until you get the output you expect.

View allows you to save your Make results to your own external file. This would allow you to take your Make results, make additional modifications, and then bring the modified file directly into Generate. You can do this if you have requirements that cannot be directly supported by the Make process.

Generate

When you select generate, checks will be created and written to an intermediate file which will be sent to the standard X9Assist Generate function. You will be presented with the Generate panel where you can specify your generate parameters to create a new output file. Generate allows configurations to be saved, which makes that process configurable and reusable similar to Make.

Generate

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Generate is a powerful data generation tool that allows you to create files from scratch to meet your specific testing requirements. File creation is typically a two step process, which consists of Make followed by Generate. Although these are often discussed as separate functions and topics, they are integrated into our Make function allowing them to be viewed and run as a single tool.

Make and Generate provide very different functions:

- **Make** is used to create items.
- **Generate** is used to wrap those items with the header and trailer records that are needed to build a valid file. For x9, this also consists of optionally drawing images that match the item level data, allowing CAR/LAR to be be run against the constructed file.

Remember that the input to Generate should be viewed as individual items which can be either debits or credits. Generate has a large number of parameters which allow you to tailor the data generation process to your specific requirements. We would like to hear from you about how this function can be improved. Your suggestions are welcome.

Generate Configurations

Over a period of time and as part of your testing processes, you will most certainly use Generate to manufacture different test data for varying purposes. Each of these may require different parameters to allow you to assign the appropriate data for the file header records, cash letter header records, and so forth.

Generate simplifies this process by allowing you to define and then save a Generate configuration. You can then restore and reuse this configuration in the future. This eliminates the need to repetitively enter information and also eliminates the potential for errors and omissions.

Input to Generate

Although Generate is design to work in combination with Make, it can also function independently. You can run Generate directly from the toolbar. When you do so, you will receive a prompt screen where you will identify the input file that should be used by Generate. This must be an Excel file (XLS or CSV) that contains the logical representation of your data records.

Generate will accept file (x9 and ach) and cash letter header (x9) records when they are present on the input CSV file. If you provide these record types, they will be used exactly as presented for file generation. If these record types are not present on the CSV input file, Generate instead creates these control records from the configuration parameters.

There are three types of inputs that can be processed by Generate:

Generate	Applies to X9 ?	Applies to ACH ?	Description
Output from Make.	Yes	Yes	Items from Make flow into Generate.
Output creation by your own internally developed applications.	Yes	Yes	You can develop processes which create this same file format and then feed that input directly into Generate. This scenario is used when you want to build your own item based data generation system, and you want to leverage the Generate capabilities as part of your process.
CSV files which are essentially in export format, structured as records and fields.	Yes	No	CSV files that are specifically contain type 25 records (checks). If you are creating a deposit, then this file can also contain one or more type 61 credit reconciliation records. If type 61 records are present, they must precede their respective checks. The amounts on the type 61 credit reconciliation records can be specified as zero, allowing X9Assist to automatically calculate the credit amounts from the debits that follow. If you specify the credit amount on your type 61 record (it is not zero), then your amount will be used. You can have more than one type 61 record on your Excel file. When type 61 records are present, there must be at least one check that follows each credit. This check file can also optionally include a file header (type 01), cash letter header(s) (type 10), and the various check addendum records; when present, these user supplied record types will be used by Generate instead of being artificially created.

New Batches / Bundles

Items will begin in a new bundle in one of several ways:

- The item count per bundle is exceeded (as defined in your generator).
- The item is a credit, and your generator indicates that credits should begin in a new bundle.
- You populate values in Bundle Cycle Number, Bundle User Field, or both of these columns. A new bundle is started automatically when these columns are populated and the value changes.
- You forcibly indicate that an item should begin in a new bundle. You can do this using the debit/credit indicator column, where you instead insert the keyword “Bundle” on a row by

itself (since this row is a separator and will not contain an actual item). Doing so will force the subsequent item to begin in a new bundle.
- You insert a “Y” in the new bundle column as identified by your reformatter.

Dates

Generate utilizes various different dates when creating output x9 files. Default values are assigned to each of these dates, and you can also modify these assignments based on your requirements. The dates assignments are as follows:

- File create date/time (default value is assigned as current date/time).
- Business date (default value is assigned as a Mon-Fri that is one day before the current date).
- Secondary endorsement date (default value is assigned as a Mon-Fri that is one day before the business date).
- BOFD endorsement date (default value is assigned as a Mon-Fri that is one day before the secondary endorsement date).
- Beginning and ending dates for created check images. Dates are assigned sequentially (within your provided range) as virtual check images are created.

These above four dates are all currently exposed and modifiable through the Generate user interface, on the dates tab. For example, the type 50 record is generated with the BOFD endorsement date, and the type 52 is also generated with the BOFD endorsement date.

Generate – X9

Generate has a standard set of columns that is utilized by Make and can also be constructed by client specific applications. Be advised that this list of fields may change from release to release, so if you create an application that uses this interface, you should design it in such a manner that it can be updated in the future should changes be needed. Please reference the notes panel that is provided by Generate (from the tool bar) for a complete list of these fields.

Generate looks at the incoming Excel file and determines which of these three file types it is being presented with. This is determined based on the record types that are on the file and the number of fields (columns) that are actually present in the data. You do not need to tell Generate what type of file you have. But you do need to ensure that it follows the rules associated with these three file requirements.

Generate will create all of the items (checks) that are routed to it for processing. Your decision as to how many checks to generate must be made by the tool that you are using to create your input to Generate. For example, within Make, you can specify how many items that you want created which will control the number of checks that will ultimately be on the x9 file from Generate.

Obviously you can use other tools to create your Generate input file. The Generate input can come from the X9Assist Export function, but can also come from your tools that manufacture your own

test cases or use Export output to build consolidated files that represent your specific testing requirements.

When Generate first starts, it will read the input file and extract the check information that is present. It will also assign default values to any data that is missing. For example, it must do this if you are providing an ICL extract since there will be several ICLR based fields that are not present in your input.

Generate will inform you of the number of checks it has found on the input file and the total dollar amount. You can double check this against your expectations before you actually generate your x9 file.

Optional CSV Record Types – X9

Generate allows you to provide certain record types on your incoming CSV file and can then optionally use these record types when the new file is constructed.

- File and cash letter headers
- Check addendum records
- Return addendum records

Use of this facility can allow you to utilize Generate for more complex situations and not have to revert to writing an application that feeds import. For example, if you need to use specific file and cash letter header records in your output file, you can cut and paste these records into your CSV file and then have Generate use those records. You can similarly do this with the check and return addendum records.

Use of these record types makes it possible to take Export output and then use that to easily create new files. You can export an existing file, remove various records from that file, and then have Generate use the resulting file with your defined Generate configuration.

Sample CSV Input File – X9

This is a sample CSV input file to generate that will create a deposit using a type 61 credit reconciliation format. The credit format (Metavante or DSTU) must be selected via the Generate parameters. Note that the credit amount is zero so it will be automatically calculated by X9Assist based on the credits that follow.

```
61,1234,"",555555550,"123456789/",0,396408128,"G","",3,"","",""
25,"","",08777026,9,"97937649/8081",0000010000,44000000,"G",0,1,"Y",03,0,"A"
25,"","",02777001,2,"29801918/8856",0000010002,44000001,"G",1,1,"Y",03,1,"B"
25,"","",09777035,6,"22690809/1921",0000010004,44000002,"G",2,1,"Y",03,2,"C"
25,"","",09777068,6,"17789149/6202",0000010006,44000003,"G",3,1,"Y",03,3,"D"
25,"","",08777050,2,"67626401/5455",0000010008,44000004,"G",4,1,"Y",03,0,"E"
```

Generation Parameters – X9

Generate parameters allow the creation of both x9 and ach files. These parameters control the batching of items and the values that are populated in file headers and trailers. Generate related parameters are provided on the “right” side tabs of the Make panel.

Generate parameters include the following:

- All file header field values can be assigned
- Credits can be automatically inserted to offset debits
- Addenda can be created
- Bundle size can be controlled
- Etc

Images – X9

Generate will dynamically create images from one of several available options:

Create missing image documents	Images will be created from your defined “missing image” document. Note that this document can be customized as part of your X9Assist installation. Creating missing image documents is the fastest image generation option. This is because individual images do not need to be drawn and then converted into TIFF format (which is the most time consuming part of the image generation process). If you are doing repetitive x9 file creation as part of a testing process, you can consider initially inserting missing images until the x9 file is being created per your requirements. Once the x9 file is correct, you can then do the more time consuming process of generating individual images.
Create missing image documents that include MICR lines	Images will be created from your defined “missing image” document. Note that the missing image document can be customized as part of your X9Assist installation. The created missing image documents will include a MICR line that is drawn from the x9 data.
Draw images using the generated MICR and amount information for the check	Images will be drawn using the generated information that is being assigned to the type 25 check detail record. X9Assist will ensure that the image meets tiff standards. You can

	customize information regarding the drawn image on the “draft check” tab within program options.

Image Templates – X9

The Image Templates tab is used to select the check artwork that will be used for image generation. There are various categories that are used to store templates for image generation:

- business
- credits
- rcc
- retail

If a check has an Aux OnUs field, then Generate will attempt to use a template from the business category. If no templates have been selected from the business category, then Generate will fall back and utilize a retail template.

Addition points for consideration:

- Available formats are defined in folder / x9_assist / xml / templates /. This folder contains file “templates.xml” which is an index of all available formats by category.
- You can add formats to templates.xml to define your own check formats. These definitions include the fonts and point sizes that are used to dynamically draw check images.

Transformations – ACH

Addenda transformations allow existing items files to be into automatically converted into more complex returns and notification of change files. This functionality will create new addendas based on your provided parameters. Using the transformation drop down, you can create files for returns, notifications of change, dishonored returns, contested dishonored returns, and refused notifications of change.

These transformations are purposefully designed to be as simple to use as possible. Via the transformation panel, you can provide a list of values in a CSV format, separated by commas) which will be applied to the generated addenda. These values are used on a circular (round-robin) basis, allowing specific data to be assigned within the created addendas. However, if you omit the transformation data, then generate will instead create the addendas using a variety of randomly assigned reason codes, change, codes, and information strings. This random assignment allows you to create a complex file in seconds.

When you first begin using the transformation function, you should omit the transformation data and allow generate to automatically construct the addenda records. This core functionality may be all that you need. For more advanced requirements, you can optionally assign the transformation data to be used, which will be saved with your generator for future reuse. The most basic value to be assigned is the return or change reason code(return reason, change reason, dishonored reason,

contested reason, or refused change reason). This is a three character value ("Rxx" or "Cxx") and applies to all transformations. This can be entered as a single value (eg, R01) or a list of values (R01,R02,R03). There are optionally more complex data formats, which are defined as follows:

- NOC data is formatted as "changeCode|changeData".
- Dishonored returns data is formatted as "dishoneredReason|returnReason|addendaInformation".
- Contested returns data is formatted as "contestedReason|dishoneredReason|returnReason".
- Refused notification of change data is formatted as "refusedCode|changeData".

These expanded data formats are optional and allow you to further tailor the constucted data to your specific requirements.

Although these system generated transformations are quick and easy, they are also a bit brute force in nature and do not always provide the granularity needed to satisfy more complex use case requirements. In those situations, you should instead consider using our Make function, where spreadsheet input can be provided on an item by item basis. Using Make is a much more detailed approach, where you fully define all contents, and hence allows you to take complete control over the entire data assignment process for each addenda record.

Generation

Use the “Generate” button to initiate file generation. This can be a lengthy process if you are drawing a large number of x9 images. Once the output file is created, you will be placed in the viewer so you can browse results. You can also save the generated file for external use, allowing you to send files to your capture and downstream applications.

Compare

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Compare is used to compare record and image data across two x9 files, subject to possible field exclusions. The currently loaded file is compared against a secondary file which is user selected. This is an excellent regression testing tool since it confirms that a modified file is logically equal to an original file, which could be the current file before modifications or a previously verified file which contains expected results.

Compare requires that the items within the file be structured in the same logical order to allow the matching process to remain aligned within the two files. Positioning is internally maintained by record type and then additionally on item amount and item sequence number. There are no other assumptions or requirements for the matching process. Any missing or inserted items will be reported and optionally listed.

The compare process reports on field level difference by field name when there are individual records that are aligned by the matching process. This makes it easier to identify the actual differences between two records that have been matched. The compare process also reports on records determined to be inserted on one side or the other of the matching process. These records are present on file one but are not present on file two.

Reviewing Differences

The compare process is invoked as follows:

- Use the 'select' button to select the secondary file to be used for the compare. Once a file is selected, it will be located and summary information will be provided which helps you determine if you want to continue with the compare.
- After statistics are confirmed, then use the 'compare' button to initiate the actual file compare process. Differences will be displayed as identified. The progress bar will be updated as the compare continues. A text file will be written to the TEMP folder and can be used (as needed) when you require a log of the identified differences.
- Once the compare has been completed, the “differences' button can be used to obtain a summary of all fields across all record types which were determined to be different.
- The “launch' button can be used to invoke a more detailed file comparison (at the record level) when a compare tool is available for your environment. This process can be very informative since it provides a better visual comparison of the differences between the two files. This comparison includes image lengths but not the actual image data. The compare tool itself is pluggable and can be modified based on your specific requirements.
- Use the 'reset' button to reset and allow the currently loaded file to be compared to another secondary file.

Excluding Fields

The default is to compare all fields within all record types. There are times when that is not appropriate. For example, you may have modified an application that creates files and you know that as a result of that change you are expecting a certain field to have different contents. In this situation, it is very helpful to explicitly exclude that one field from the comparison so you can then easily determine that all other fields within the file are equal as expected. This exclusion is accomplished by using the selection panel on the right. One or more fields can be excluded from the compare process. You should indicate the fields to be excluded before you select the secondary file to be compared.

Implementing an External Compare Tool

The file compare process can be further enhanced by adding an external compare tool. This compare can then be launched after the file compare has been completed and allows you to do further analysis of any differences which may exist. This is an optional process to add value to your environment. This is a text-to-text comparison of two text files which are created during the compare process. Launch of the external compare tool allows you to visually explore the differences between the two files that have been compared. This is possible due to the powerful string matching capabilities of the text compare tools that are available in the industry today.

X9Assist is not hard-wired to a specific external comparison tool. You can implement the preferred external comparison tool of your choice through the “comparex9.bat” batch script. This single batch script is used to both identify and launch the comparison tool. This batch script allows the compare tool to be generic and external to our comparison process.

Our distribution by default supports the following external compare tools:

1) WinMerge: a freeware Windows alternative.
2) Meld: a freeware alternative which is available for Windows and Linux.
3) UltraCompare: a powerful file edit / compare tool and an industry leader which is installed by many organizations on a world wide basis.

Batch file “comparex9.bat” is located in the program launch folder. A parameter of “-isDefined” is initially invoked to determine if the compare tool exists. This pre-launch must return an exit code of 10 when the compare tool exists. The compare is then launched using this same batch file, using two parameters which represent the two files to be compared. The comparex9.bat file can be modified to invoke the preferred tool for your environment.

WinMerge Usage

The following options should be set to maximize your usage.

1) View from the menu bar
   a) Enable line differences
   b) Enable line numbers

   c) Enable margins
2) Options (wrench) from the tool bar
   a) General – enable compare white space
   b) General – enable ignore case
   c) Editor – enable view line differences at character level

Meld Usage

The “up arrow” and “down arrow” on the tool bar can be used to skip to each identified difference within the compare.

UltraCompare Usage

The “up arrow” and “down arrow” on the tool bar can be used to skip to each identified difference within the compare.

Clone

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Clone is used to replicate the items within the currently loaded file using a defined replication factor. This function is extremely useful when you want to create a larger file, where replicated items are satisfactory for your testing purposes. An example would be the need for much larger files for stress testing purposes. Note that it is also possible to use clone to build a much bigger file that is then scrubbed. This dual process can be used to quickly generate a very large file that can be used for system stress testing.

Clone will automatically renumber the bundles within the file to ensure that the bundle identifiers are unique per x9 and ach rules.

You can specify the number of times that each bundle will be duplicated. The new bundles will be appended to the end of your existing file data.

For x9 files, if the currently loaded file contains a credit reconciliation record, then you can also indicate how you want the resulting cloned output to be constructed. You can either duplicate the input which will create multiple credits, or you can combine the check volume into the credit reconciliation record that already exists.

Clone will launch a new validation process for the cloned file. The cloned file remains in memory after the cloning process has been run. You must save the results to an external file when you have reviewed results and determined that the output is beneficial based on your requirements.

Merge

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Merge allows the content of one or more additional user selected files to be appended to the end of the currently loaded file. Merge is a convenient way to combine multiple files into a single (larger) file. This is helpful in several different scenarios:

- When you want to combine various files with different test cases into a single file that can be used for ongoing testing.
- When you want to combine multiple files to create a single larger file, possibly to be used for stress testing purposes.
- When you purposefully want to create invalid data conditions by combining data of unlike transaction types within the same logical file.

Merge can be run repetitively to include more and more data into a single file. This is helpful when you are pulling in files from multiple input folders. When doing multiple merges, you must save the results each time to an external file before proceeding to the next merge step.

Merge will automatically renumber the bundles (batches) as needed to ensure that their individual identifiers are unique.

For x9 merge, you have the option to either merge as bundles or merge as cash letter(s). When you merge as bundles, the bundles from the selected x9 file will be appended to the last cash letter that exists within the currently loaded x9 file.

    - When you merge as cash letters, the cash letter(s) from the selected x9 file will be appended to the end of the currently loaded x9 file.
    - When merging as bundles, you have the additional option to copy the origination / destination RT from the cash letter header record to the merged bundles. This is a validation requirement per UCD rules. This option will only be applied when merging bundles is selected.

Regardless of which merge process is selected, you have a common option to assign new item sequence numbers to all output items. When this option is selected, new sequence numbers are assigned to both the current and the inserted (merged) items. Selecting this option will ensure that your new x9 file does not have duplicate item sequence numbers.

Selecting Files

You can select multiple files as input to a single merge operation. This is done in a standard way based on your OS platform. For example:

- You can select multiple files by holding down the CTRL key as you select individual files.

- You can select a range of files by selecting the first file within the range and then holding down both the CTRL and SHIFT keys as you select the last file.
- You can select all files within a folder by using the CTRL A key combination.

Your selected files will be displayed in the “Selected Files” panel on the right.

Identifying the Output File

Merge by default will write to an intermediate file which is located in the temp folder. This default action is appropriate since it is important that you closely review the created output file before you save it to your target file. However, in some situations, it can be beneficial to bypass the writing of the intermediate file and instead write the merged output directly to you target output file. Doing so saves a step which can potentially take a measurable amount of time, and especially when results are being written to a networked drive.

In support of this optimization, Merge allows you to optionally choose your target file (before the merge process is initiated) thus eliminating creation of the intermediate file. When doing so, the target file will still be reread and validated on completion. You should closely review the created file and ensure it is correct based on your requirements and before it is used for any subsequent purpose.

Initiating the Merge

Once you have selected files, press the Merge button to initiate the merge process.

Merge will launch a newly created file validation process for the resulting merged file. It is very important to review the validation results to ensure that the created file does not have any unexpected errors. The merged file remains in memory after the merge process has been run. If the output was written to the intermediate file, then you must save these results to an external file once you have determined that the results are corrected and beneficial per your requirements.

Repackage

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Repackage allows the batches/bundles within the currently loaded file to be rebuilt and the items within these bundles to be optionally reordered. This tool is especially helpful after merge, when merging multiple small files, since it allows the resulting bundles to be combined. It is also useful when you have an x9 file with multiple cash letters, since the output will always be a single cash letter. Another purpose is for test environments, when it would be helpful for your items to be in ascending amount order.

Repackage allows you to specify the number of items to be placed within each bundle. For example, you can take a file that has 500 items per bundle and create a new file that has 300 items per bundle.

The sequence of items (debits and credits) can be selected as follows:

- As is (items will remain in the same order as the input file).
- Amount (items will be reordered to be in ascending amount sequence).
- Item sequence number (items will be reordered to be ascending on item sequence number).

The attributes to be assigned to the created bundle records can be entered prior to initiating the repackaging operation. These parameters are automatically extracted from the first bundle record within the x9 file, so they will typically not require any modification on your part.

Items are written with all of their associated addendum and image records exactly as present on the input file. There are no changes to any of those record types.

Trailer records will be updated appropriately based on the consolidation actions which have been taken. This will result in updates to all trailer record types within the file.

For ACH, all standard entry classes are supported with their respective addenda formats.

For X9, ICL and ICLR files types are supported with the following additions:

- Credit reconciliation records (type 61) within an ICL file.
- User records (type 68) within an ICL file.
- Box summary records (type 75) within an ICL file.
- Routing number summary records (type 85) within an ICL file.
- The output file will always contain a single cash letter.

Convert X9 to ACH

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**NO**	**YES**	**NO**

Convert X9 to ACH is used to create a new ACH file from the currently loaded x9 file. A user selected standard entry class is used to build the new ACH file. This includes the various entry classes that represent checks (ARC, BOC, POP, RCK, TRC, and XCK) as well as some more generic entry classes which may provide utility within the user environment (CCD and PPD).

Conversion Overview

Convert is a generalized conversion tool that can be used to translate from one file format to another. X9Ware is in a unique position to implement this functionality given our support for both the X9 and ACH file formats. This allows the user to load an x9 file, browse that file to ensure the content is correct, possibly modify the content of that file if needed (for example, remove records), and then use those results to create an ACH file. All of this is done within the same tools and using using a consistent user interface.

The following fields must be entered for the X9 to ACH conversion:

- Debit transaction code: defines the transaction code to be assigned to debits.
- Credit transaction code: defines the transaction code to be assigned to credits.
- Identification number: defines data that is assigned to ACH field 6.7. This value will be automatically assigned from the check serial number for standard entry classes of ARC, BOC, POP, and RCK. Other entry classes allow you to explicitly assign the value to be used. When a value is not entered and has no default, it will otherwise be assigned from the check serial number.
- Receiver information: defines data that is assigned to ACH field 6.8. This value will be automatically assigned from the process control field and the item sequence number for standard entry classes of TRC and XCK. Other entry classes allow you to explicitly assign the value to be used. When a value is not entered and has no default, it will otherwise be assigned from the trc/xck process control and item sequence number format.
- Discretionary data: is an optional two character field that is assigned to ACH field 6.9.
- Starting sequence number: is a seven (7) digit number that is used to sequentially assign the trace number for the created items, which is populated into ACH field 6.11. The overall trace number is 15 digits and consists of the 8 digit ODFI routing number which is suffixed by this sequence number. On a default basis, this starting sequence number is “DD00000” where DD is a variation taken from Julian day within the current year. However, this default would not be appropriate for large files and when multiple conversions are done

within the same day. In those situations, the user must provide the starting sequence number which guarantees the trace number to be unique.

These various parameters will be written to user preferences at the end of the current user session, allowing them to be easily restored and reused in your next session.

Generators

Convert/Generate is an integrated process where the generator is accessible during the conversion process. Convert allows a generator to be loaded, modified, saved, and ultimately used to create the output file. This process is extremely integrated. Generator definitions are in a common format that is shared with Make.

The “generate” tab defines all of the file level parameters that are available. These parameters represent a generate configuration that can be saved to xml and then repetitively used in the future. This reuse allows you to ensure that the created files are always generated using the same parameters and thus provide the same results.

Running Convert

Use the “convert” button to initiate the file conversion process. Convert will launch a new validation process for the converted file, which remains in memory after the conversion process has been run. You must save the results to an external file when you have reviewed results and determined that the output is beneficial based on your requirements.

Standard Entry Classes

The ACH file format has certain standard entry classes which represent a paper check item, and those formats are especially appropriate for these actions. These are ARC, BOC, POP, RCK, and TRX. Additionally, we support CCD and PPD since they are common layouts; these may be used for data exchange within an organization, but should not be used to route the resulting transactions on an external basis. The supported standard entry classes are:

- ARC, which represents an eligible check received via the U.S. mail or at a drop box location, which was presented for the payment of goods or services. The source document (the check) is used to identify the Receiver’s routing number, account number, check serial number, and dollar amount for the transaction.
- BOC, which represents an eligible check received at the point of purchase or manned bill payment location for the in person purchase of goods or services. The source document (the check) is used to identify the Receiver’s routing number, account number, check serial number, and dollar amount for the transaction.
- POP, which represents a payment for the in person purchase of goods or services by Receivers. These debit entries are initiated by the Originator based on a written authorization between the Originator and Receiver and notice provided by the Originator at the point of purchase or manned bill payment location. The source document, which is

voided by the merchant and returned to the Receiver at the point of purchase, is used to collect the Receiver's routing number, account number, and check serial number that will be used to generate the debit entry to the Receiver's account. This type of entry may only be used for non-recurring, in-person (at the point-of-purchase) entries for which there is no standing authorization with the merchant for the origination of ACH entries to the Receiver's account.

- RCK, which represents a debit initiated by an Originator pursuant to an oral authorization obtained over the telephone to effect a transfer of funds from a Consumer Account of the Receiver. This entry may only be used when there is no standing authorization for the origination of ACH entries to the Receiver's account. A TEL entry would instead be used when there is an Existing Relationship between the Originator and the Receiver, or, when there is not an Existing Relationship between the Originator and the Receiver, when the Receiver initiates the telephone call.
- TRC, which represents truncated checks being safe kept by the keeper bank (Originator) as defined by a check truncation program. This entry has one transaction per truncated check. The TRX format (which is not currently supported by this conversion process) allows financial institutions to use a single transaction entry to carry information from multiple checks, where each check is identified within addenda records attached to the item.
- CCD, which represents funds are transferred between unrelated corporate entities, or transmitted as intra-company cash concentration and disbursement transactions. This entry class can be used for standalone funds transfer, or it can support a limited amount of payment related data with the funds transfer. The CCD class is not appropriate for these items, but is included to facilitate data conversions for files processed within a given single environment.
- PPD, which Direct Deposit or Pre-authorized Bill Payment. Direct deposit is a credit application that transfers funds into a consumer's account at the Receiving Depository Financial Institution. The funds being deposited can represent a variety of products, such as payroll, interest, pension, dividends, etc. Pre-authorized payment is a debit application. Companies with billing operations may participate in the ACH through the electronic transfer (direct debit) of bill payment entries. Through standing authorizations, the consumer grants the company authority to initiate periodic charges to his account as bills become due. This concept has met with appreciable success in situations where the recurring bills are regular and do not vary in amount insurance premiums, mortgage payments, and installment loan payments being the most prominent examples. Standing authorizations have also been successful for bills where the amount does vary, such as utility payments. The PPD class is not appropriate for these items, but is included to facilitate data conversions for files processed within a given single environment.

Add Items to PickList

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Add Items to PickList allows items within the currently loaded file to be added to the PickList, which is an internal filter used to collect specific items of your selection. This function can be invoked from the menu bar (under the Filters menu) or by using a button on the tool bar.

The PickList is implemented as a system filter, just like the other filters that are created by such functions as Find and Search. This allows the items within the PickList viewer to be browsed using the filter that is displayed within the DashBoard. The filter viewer can also be used to remove items from the PickList when needed.

Before you invoke Add Items to the PickList filter, you must select the item (or items) to be added within the viewer. This selection can be:

- An item (debit or credit)
- A batch / bundle
- An entire cash letter.

A pop up message will be displayed when the positioning is incorrect based on this criteria.

Individual items can only appear once within the PickList. You will receive an error beep if you attempt to add the same item more than once to the PickList.

Once you have added items to the PickList, you can then do other functions such as:

- Print a table of the items.
- Print images associated with the items.
- Create a new file from the selected items.
- Merge the PickList with other filters to then allow you to then perform the above functions.

Summary

Add Items to PickList allows you to individually select the items to be added to the system PickList. Once you have accumulated all items in the PickList, you can then use it as the basis for other reporting and file creation functions.

Returns

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**NO**	**NO**

Returns allows Image Replacement Documents (IRDs) to be decisioned with a user supplied return reason, either individually or in bulk. Decisioned items are added to a user constructed returns list, that can be subsequently reviewed and use as a filter to actual print the IRDs. Returns is typically used as a mechanism to print a selected number of items from an ICL.

IRD Types

IRD items are entered as one of two types:

- An IRD Substitute is a negotiable instrument used to represent the digital reproduction of an original paper check. Substitute checks are recognized as legal checks as long as the instruments meet specific requirements. These requirements include the faithful reproduction of the paper check and warranty of the instrument by the "reconverting bank"—the financial institution that created the substitute check or the first financial institution that transferred or presented it during the check clearing process. Substitute checks are also subject to the UCC, existing federal and state check laws, and regulations specific to consumer rights that affect the acceptance of these instruments. Although a substitute check is subject to the UCC and existing state and federal check laws, the Check 21 Act takes precedence over these other laws and regulations for this instrument.
- An IRD Return represents an item that was not paid by the issuing bank for some defined return reason (eg, insufficient funds). In this situation, the paying bank stamps the item as NSF (non-sufficient funds) which identifies the reason for the return. When an item is being returned, a “5" will be placed in the MICR line EPC field to identify the substitute check according to ANS X9.90, along with the routing number of the depository financial institution and the dollar amount of the substitute check. The paying bank returns the unpaid returned item through the routing process to the BOFD for exception handling, which would typically include debiting the customer for the returned item, sending a charge back notice to its customer (either electronically or through the mail), and most probably assessing a fee which is associated with the returned item process.

Adding Items to the Returns List

Items are added to the Returns List through a series of one or more user interactions. The ICL viewer tree (on the left) is used to select items to be added. This can be done in multiple ways:

- Items can be added one at a time. This is done by selecting the item in the viewer tree and then invoking Returns from the tool bar.

- Multiple items can be added in a single interaction. This is done by holding done the CTRL key, selecting the items to be added using the mouse, and then invoking Returns from the tool bar.
- A block of items can be added in a single interaction. This is done by selecting the first item within the block, holding done the CTRL/SHIFT keys and selecting the last item within the block, and then invoking Returns from the tool bar.
- An entire bundle of items can be selected in a single interaction. This is done by selecting the bundle and then invoking Returns from the tool bar.
- An entire cash letter of items can be selected in a single interaction. This is done by selecting the bundle and then invoking Returns from the tool bar.
- An entire file of items can be selected in a single interaction. This is done by selecting the bundle and then invoking Returns from the tool bar.

The selection process can be used repeatedly to build the Returns List with the desired items. The accumulated items will be displayed within a system filter. Items can be added or removed from the list as necessary. Items can also be reviewed and launched using the filter list. Once the content of an Returns List has been confirmed, the list can then be used as input to Print Image.

Entering IRD Information for Individual Items

The Returns panel allows you to enter information that is associated with each IRD within the overall list. The following fields must be entered for each IRD:

- Creator routing number – which identifies the financial institution that is creating this image return document.
- Business date – the date that the IRD is printed.
- Item sequence number – the item sequence number that is associated with the IRD.
- Return reason – the return reason (for returned items).

Items are typically decisioned one at a time, which allows the appropriate return reason to be assigned to each item. Alternatively, items can be processed in a grouped (repetitive) manner, which is appropriate when the same return reason applies to all items, and the default action of an incremented item sequence number is acceptable.

The number of remaining items to be decisioned will always displayed on the action line in the lower left corner. Buttons on the action panel are used to either apply (add) or apply all items to the accumulated IRD print list. The process continues for all remaining items within the list.

Once all selected items have been decision, the entry process will be terminated and the resulting Returns List will be displayed. You can use the filter tab to launch to any given item or alternatively remove items that were added in error. Also note that the return reason is displayed for each item within the filter list, which allow you to review the assigned returned reasons for all items.

Using the Returns List to Print Item Images

Once the Returns List has been finalized and confirmed, it is then used as a filter for other application functions including Print Images.

Print Image supports various page layouts, and very specifically supports 3x1 IRD format which is used to print IRD documents. In addition to standard layouts, Print Image allows custom formats to be defined when your page (paper) layout varies from our standard format.

When printing images, our "**Letter 3 x 1 IRD Print**" format should be applicable for most user environments. This print format can be adjusted based on your check template and specific printer environment. Contact us if you need assistance to implement your specific check layout. The important attributes of this image print format are:

- Letter.
- 3x1 format.
- Front and back images.
- Two sided printing (front images on even pages and back images on odd pages).
- Collated which retains the item order after printing and bursting.
- 1200 DPI to improve document quality – should be set which will trigger the print image function to use 1200 dpi whenever possible.
- Redraw MICR from x9 data – should be specified since it allows the MICR line to be drawn on the IRD using the E13B font.
- Return MICR 8 digit routings in xxxx-xxxx format – should be specified to insert the dash into the routing field when an eight digit routing is present.

Merge Filters

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Merge Filters allows you to merge two or more filters into a single filter which will contain all items that existed across the selected filters. A filter is essentially a list of items within the currently loaded file. You must have at least two filters currently defined before you invoke this function. Once the merge filters dialog is displayed, you then must select at least two filters that are to be merged.

Filters can be created in multiple ways, with the easiest being the use of the Find and Search functions. You can create a list of filtered items directly from these search tools. However, filters can also be created using the PickList tool, where you can add individual items to a filter based on your specific selection requirements.

The resulting items will be stored in the first selected filter and the subsequent filters will be removed. Items in the merged filter are stored ascending by their record number position. This means that the items are logically inserted at their appropriate positions based on their record sequence number. Items can only exist once within the resulting filter; any duplicates will be identified and will only exist once in the merged filter.

The resulting filter can be used by various other functions such as create file, print, print images, delete items, and merge filters (again).

Create

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Create can be used to manufacture a new file using one or items from the currently loaded file, typically from a list of filtered items. Input to this process can be either a filter that contains the specific items to be included in the new file, or alternatively the entire content of the current file. Filters may be created using the PickList, Find, Search, and Error Summary functions.

A common use of this tool is to use PickList to select individual items from the currently loaded file and to then create a new file directly from those specific items. In addition to manually selecting items, the system generated filters can be used to build files as well. For example, you can validate a file and then use the Error Summary tab to create a filter of just those items that have a specific error. You can then create an output file that consists solely of those items.

For x9, the output file can be either ICL (forward presentment) or ICLR (returns). What is particularly useful is that Create can be used to transform an ICL file to ICLR and similarly transform an ICLR file back to ICL. These conversions are especially helpful during various on-boarding and system testing scenarios.

You can create the new file directly from the selected items or you can instead route the items through the Generate function. If you create the file on a direct basis, then the selected items will be written exactly as they appear within the current file (including the attached addenda and image records) and will be written in their current format (ICL or ICLR). Items will remain within their current file and bundle structure. Repackage can be used to further consolidate the bundles when that is specifically desired.

You can alternatively create the new file through the Generate function. When this option is selected, the items will be reformatted from into CSV and then routed into Generate for additional reformatting. When using this approach, the attributes and reformatting of these items is then completely controlled by a Generate Configuration which is user defined and can be saved for repetitive reuse.

For x9, when using Generate you can transform items from ICL to ICLR or from ICLR to ICL. You also have complete control of the file header, cash control header, and bundling of these items.

Input Source

Input items are taken from the currently loaded file. When creating the output file using Generate, you can select the input source as either the entire file (all items) or only those items that are associated with a defined filter.

X9 File Creation Method

Generate From Filter requires that you select the method that will be used to create the new x9 file. One of two selections must be made:

1. Create a new file directly from the existing record data. Records are written in the existing format and the data is not modified in any manner.
2. Reformat and process through Generate. When this option is selected, the Generate button will take you to the Generate panel, where you can review and modify your generate parameters before you initiate the file generation process. You can also use this interface to specify and save a customized generator that can be repetitively reused in future user sessions. When used in this manner, you can quickly do functions such as:

    - Use an ICL to create an ICL with different file headers
    - Use an ICL to create an ICLR with the same items and possibly different headers

Routing Items through Generate

When items are routed through Generate, the File Control header, Cash Letter header, and Item Addenda records are normally retained when using this tool. However, options are provided to drop these record types on the CSV that is sent to Generate to provide additional flexibility by allowing Generate to take full control of those record types. These options would typically be used in a testing environment to meet specific data needs.

- Eliminating the header records within the intermediate CSV file, allowing data on the File Control, Cash Letter, and Dates tabs within Generate to create those record types.

- Eliminating the addendum records within the intermediate CSV file, allowing the Addenda tab within Generate to alternatively create those record types.

Mixed Debit-Credit Files

Create allows both debits and credits to be selected. The processing varies a bit by dialect:

- X9 files typically consist of checks which might be offset by a single credit. In this scenario, the credit can either be positioned before or after the debits, by most typically appears first. When using the PickList, Create allows you to optionally select this credit, to be included in the output file. Create will then ask you if the credit should be modified to match the debit total of the selected items.
- ACH and CPA005 files more typically consist of both debits and credits. In some situations, a file may consist of debits offset by a single credit, or credits offset by a single debit. In this case, Create recognizes that there is a single offset transaction and will ask if the amount on that balancing item should be modified.

Summary

Create is a powerful tool that can be used to create new files from specific items of your selection. Using the interface to Generate, you can also customize the file header information and other parameters associated with the newly created file. These parameters can be saved in a generator that can be referenced and invoked in future user sessions.

Amend

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Amend is a low-level tool that can be used to manufacture test files for complex situations, where there are specific requirements that go beyond the capabilities of Make/Generate. Amend allows you to manipulate files at the record and field level, in ways that were not previously possible. This is an advanced tool that can be used to do such operations as:

- Copy and paste a single record or group of records from one place to another within a file.
- Replicate a single record or group of records within the a file.
- Delete a single record or group of records within a file.
- Easily find occurrences of specific field values and replace them with other values.
- Easily manipulate fields in more complex ways than X9Assist modify or search/replace.

All of this is helpful when you are facing a difficult file manipulation task. This is especially true when you need to create a file for “negative testing”, where the file must contain some invalid data scenario that Make/Generate cannot create for you. This situation arise because Make/Generate is geared towards creating valid files. Amend, however, can further manipulate a file that is created using Make/Generate to meet virtually any specific testing requirement.

Amend is built on top of several of our other tools, as follows:

- Export is used to convert the currently loaded file to a CSV, in our internal work folder.
- Our CSV Editor is used to manipulate the CSV file. Our CSV Editor enables this entire process, since it supports a variable number of fields (columns) on each row. This is extremely important, since most CSV editors will force a saved file to have the same number of columns on each row. Although this might be reasonable in most situations, in our case this is not acceptable, since rows can represent different records types, where each can have their own unique field count. Our CSV Editor fills this gap, since it preserves column count at the row level when a file is saved. In order to use Amend, you will need to become fluent in the usage of our CSV Editor.
- Import is used to read the CSV file that is saved within the CSV Editor and ultimately recreate a modified version of the original file. This new file will be validated, using our standard validation rules, and can be further inspected to ensure your modifications are as you would expect based on the changes that have been made.
- Finally, Save can be used to write the modified file to an external file of your choosing.

Summary

Amend is a powerful tool that is typically used to manipulate a file in ways that cannot be achieved using our more standard tools such as Make/Generate, Modify, or Search/Replace. As with all complex tools, it should be used carefully with all output verified for correctness.

Reversals

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Reversals can be used to create a new reversals file using one or items from the currently loaded file, typically from a list of filtered items. Input to this process can be either a filter that contains the specific items to be included in the new file, or alternatively the entire content of the current file. Filters may be created using the PickList, Find, Search, and Error Summary functions.

A common use of this tool is to respond to a file that has been previously created in error. In these scenarios, an entire file will have to be reversed. This reversal process must be used extremely carefully, since you never want the reversal file to be flawed as well. This means that you need to fully understand the reversal process, and more importantly closely review the resulting output file that is created. This review needs to be done by your operations and technical teams (it needs multiple reviews and sign-offs). The review needs to look at the transactions, transaction codes, addenda records, file headers/trailers, and file totals. A careful review must always be performed.

You can create the new file directly from the selected items or you can instead route the items through the Generate function. If you create the file on a direct basis, then the selected items will be written exactly as they appear within the current file (including the attached addenda and image records). Items will remain within their current file and bundle structure.

As described above, ACH and CPA005 reversals can be either written directly or processed through Generate. The primary reason to go through Generate would be to further customize the output file (eg, change the header/trailer records). X9 files must go through Generate, since those items must have their images redrawn as debit-credit memos. In this situation, our standard check templates can be used, or you can alternatively design your own template based on your specific requirements.

Reversals files have their debits flipped to credits and alternatively credits flipped to debits. These transaction code transformations vary a bit by dialect, as follows:

- X9 is the most complicated, and especially when credits are defined as type 25 records with some type of transaction code and/or routing number. A user defined configuration table can be constructed that identifies the transaction code and/or routing number to be assigned to POD credits. This swap table can have one or more entries, where the first matching entry will be selected and applied. Reversals will be always routed through Generate, which will be used to set further output file attributes. This must include the credit record type and format, and the image templates to be used to when creating the reversal transactions.
- ACH and CPA005 reversals are much more straight forward. Most typically, an entire file will be reversed. A transaction code swap table is provided for ACH, where the default should be appropriate based on industry standards. A transaction swap table is not needed for CPA005, since reversals are identified as a separate record type.

Offset Balancing

Reversal files can consist of debits, credits, or both. In some situations, the file may consist of debits offset by a single credit, or credits offset by a single debit. In this case, the reversal process recognizes that there is a single offset transaction and will ask if the amount on that balancing item should be modified. When using a filter, if the offsetting item is not included in the filter, then reversals will ask if that item should be added. Note that our automated balancing process requires at least two items offset by a single item. These items can be be located in different batches. If the offset exists in a batch by itself, then that batch structure will be retained even when a filter is being used to identify the items to be reversed.

Summary

Reversals is a powerful tool that can be used to create new files from specific items of your selection. Using the interface to Generate, you can also customize the file header information and other parameters associated with the newly created file. These parameters can be saved in a generator that can be referenced and invoked in future user sessions.

Delete Filtered Items

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Delete Filtered Items is used to delete a list of items from the currently loaded file. These filter lists are created by other tools such as the find, search, and pick list facilities. This is a powerful function, since it allows items a group of specific items to be removed from the current file.

As a review, Find and Search can be used to create lists of filtered items. Once one or more filter lists are created, there are a series of tools that can be used against the filters. For example, you can browse the items that exist within a given filter list, remove individual items from the filter, or merge two or more filters to create a new filter. The combination of these functions enable very powerful delete operations such as:

- Search on the certain field and create a filtered list
- Remove items from that list
- Search on another field and create a second filtered list
- Add specific items to the pick list
- Merge the three lists to get a single list that can be reviewed
- Delete the items using the provided list

As an alternative to using Find / Search, you can also manually add items to an internal filter which is called the PickList. This is a special filter name, but otherwise behaves just like any other filter list and can be used by all of our filter based tools. One more more items can be selected from the viewer tree and added to the PickList in one of several manners:

One Item	Click an item in the ICL tree to select the desired item.
Multiple non-consecutive items	Click an item in the ICL tree to select the first item. Then hold down the CTRL key and click additional items to add them to the selection.
Consecutive items	Click an item in the ICL tree to select the first item in the group. Then hold down the SHIFT key and click the last item in the group.

You must have at least one filter defined before you invoke Delete Filtered Items. If there is only one filter currently defined, then it will be assumed to be the desired filter and you will immediately receive the “are you sure” message to confirm the delete. If there are multiple filters currently defined, a panel is displayed allowing you to select one or more filters which will identify the items to be deleted.

Once you have selected the filters which identify the items to be deleted, you must press the delete button to initiated the delete process. A confirmation message will be displayed which will allow you to confirm that the items are to be deleted.

The filters will be removed since they contain items which will no longer exist.

The deleted items will be added to the current modification log which can be both viewed and optionally saved when the current file is itself saved. The modification log can also be used to revert any of the deletes that were applied.

The counts and amounts in the trailer records for the current file will be automatically updated when the items are logically removed. You will not have to run repair. Entries will appear in the modification log for all trailer record updates that were applied.

After your review, the modified file must be saved to the external file of your choice.

Filter Export-Import

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Filters represent a subset of items within the currently loaded file. Any individual item can be only occur once within a filter, but can exist in separately defined filters at the same time. The items within a filter are arranged in a sequential order (relative to how they appear within the current file) regardless of the order that they may have been added.

Please read the “Filter” topic for more information regarding filters and their capabilities.

Filter export-import share a common CSV format, which means that an exported filter can be subsequently used by the filter import tool. A filter can be created using our various functionality (find, search/replace, individual item selection, etc) and then exported to an external CSV file. In more advanced situations, filters can also be created using other external tools to address user specific needs. The filter identifies individual items. It can then be subsequently imported from the exported filter content. An exported filter contains the following five (5) fields:

- Item sequence number
- Amount
- Routing
- Account number
- Check number

Typically, filters are used through several consecutive steps:

- First, build a filter using a tool such as find. This allows you to enter item selection criteria which is then matched against the currently loaded data file. The result will be a filter tab that contains the matching items.
- Second, use the filter-export tool to export the item selection criteria to a CSV file. Export always writes the output CSV file in a standard five column format (ittem sequence number, amount, routing, account number, and check number).
- Finally, at some future time, some other file can be loaded and the exported CSV criteria file can then be imported and matched against the current data file. This allows the already entered criteria to be applied again, to some other file.

However, CSV filters can be used for more expansive purposes other than just our standard tools. For example, you might have an external process that creates an account number list filter, containing just two columns (routing and account number). This could be considered to be general purpose pick-list that could be applied frequently to files, as needed. This filter could be maintained using a text editor, but could also be a dynamic list that is created by an application program. Once this list is constructed, it can be imported and then applied to any file that is loaded.

In this situation, you need to describe columns that are associated with your CSV criteria file. Our standard is to have five attribute columns, but you can have less or more than that depending on your specific requirements. Just as importantly, the columns do not have to be consecutive, which means that you might have a CSV file from some other purpose that can be imported an applied. In these situations, just remember:

- Load/Save is used to define the filter configuration, which defines the columns that are to be used for matching. Many users will not need this, since they are using our standard five column attributes definition. However, if this is not the case, then you can configure and save your own column definitions. This information is stored as an XML file in folder / xml / filter.
- Import is then used to load the CSV datea file, which is mapped based on the current filter configuration.

Remember that filters can be used for a variety of purposes:

- Create File can be used to create a new file that contains only those items that exist within a selected filter (this uses the filter list for inclusion).
- Create File can be used to create a new file that does not contain those items that exist within a selected filter (this uses the filter list for exclusion).
- Delete items from the currently loaded file.
- Image Print the selected items.

Duplicate Items Detector

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**NO**	**NO**

The Duplicate Items Detector can identify duplicate items either within the currently loaded x9 file, or across multiple x9 files. Duplicates can be identified for the following types of items:

- Check detail records (type 25)
- Return detail records (type 31)
- Credit reconciliation records (type 61)

Duplicates are considered within record type, which means (for example) that you cannot have a duplicate that consists of a debit and a credit, even if they have the same MICR line and amount.

Duplicates within the current X9 File

When you open the duplicate panel, X9Assist will automatically search for duplicates within the current x9 file (using default duplicate criteria) and will inform you of how many duplicate items were located.

You can then change the duplicate criteria as you see fit and use the “Check” button to initiate another duplicate search.

The following criteria can be included in the identification of duplicate items:

- ABA number
- OnUs field (which in most cases will include the consumer check serial number)
- Auxiliary AuxOnUs
- External Processing Code (EPC)
- Amount
- Item Sequence Number

When you have duplicates, X9Assist will display a screen that shows you both sides of each duplicate occurrence. The duplicate items panel has a divider (“==========”) between each duplicate case.

Each duplicate provides information on the key (“duplicate […]”) that was used to identify the duplicate, which is based on the criteria that you have specified. Each duplicate also provides information on the source of the duplicate item (“from file….”).

There will be two or more lines reported for each duplicate. Each line has a prefix that will tell you the file, record number, and record type that was the source of the duplicate item. Note that a single item may be duplicated with one or more other items, hence there can be two or more lines that are shown between each of the dividers.

The following buttons are available from the Duplicate Items Detector panel:

- **Check**: initiates another duplicate items analysis and is used when you have changed the criteria and want to update the results based on your new selections.

- **Filter**: creates a new filter for those items that are duplicates within the current x9 file.

- **Delete**: deletes those items that are duplicates within the current x9 file. You will be asked to confirm the delete before it is initiated. The deleted items are added to the current modification log (as supported by Modify). This will allow you to review the deleted items and revert any of the items as you so desire. You will need to run a Repair to correct trailer totals. Also note that if this file contains credits and you are deleting the debits, then the credit amount(s) will also have to be modified. Once your review of the x9 file is completed, you can save the modified file to a new x9 file name.

Duplicates Across Multiple X9 Files

The Duplicate Items Detector can be used to identify duplicates that exist across multiple x9 files. The identification of such duplicates can become a complex process and X9Assist provides tools to help in these difficult situations.

Suppose you have duplicates that involve three different files. In this example, you would do the following:

1) Use X9Assist to open what you consider to be your primary x9 file associated with this event. This is the only file that you will be able to use for the “Filter” and “Delete” functions.
2) Launch the Duplicate Detector for this file.
3) Within the Duplicate Detector, use the “open” button on the tool bar to open the second file. You will get a popup message of how many items were found and loaded from this file.
4) Within the Duplicate Detector, use the “open” button on the tool bar to open the third file. You will get a popup message of how many items were found and loaded from this file.
5) Now assign your duplicate items criteria
6) Press the “Check” button.
7) You will get a popup message that identifies the number of duplicate items that have been found.
8) If necessary, you can adjust your duplicate identification criteria and hit “Check” again.
9) As necessary, you can repeat this process.
10) You can now use the “Filter” button to allow you to generate a list of the duplicate items that exist within the first file that you opened.

Duplicate Reporting

The Duplicate Detector generates a display of all identified duplicate items so that you will know which items and which files are involved in the duplicate scenario. You can save this text file which provides great insight into the items that are participating in your duplicate event.

Duplicates within the current x9 file can be inserted in a filter. You can Use the Excel Exporter to export these items into an Excel workbook.

As you are well aware, duplicate items situations are always complex and demanding. X9Assist provides generic tools that are designed to provide help and guidance. Please let us know how we can improve upon these tools. As always, your suggestions are welcome.

Swap Images

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**NO**	**NO**

Swap Images includes tools that allows x9.37 item images to be manipulated in several unique ways. Swap is especially useful because it operates against a series of consecutive items, where you can identify the first item to be manipulated, along with the number of items to be updated. Swap operations can be run across both bundles and cash letters.

Currently supported operations are:

- Swap the front and back images within a given item. This is a simple swap that exchanges the front and back image. This swap can be used to correct a capture problem, when the assignment of front and back images was incorrect. Another possible use would be to manipulate data for testing, where you are specifically need images to be exchanges to meet the desired use case.

- Create a code line mismatch situation, where the MICR line data that is assigned in the item record does not match the front side image attached to the item. This operation can be applied to any item type (record types 25, 31, 61, or 62). This is a helpful process when you want to purposefully create image mismatch conditions to be used for application testing. The back side images are not updated during this operation – only the front side images are changed. Mismatches are created by assigning the front image of the next (subsequent) item to the current item.

Swaps are logged as modifications, which allows swap operations to be individually reverted. For example, you can swap the front-back images on an item and then revert the swap on just the front side image. By doing this, the front image would then be assigned as both front and back.

As part of swapping the actual images, the image length is also updated within the type 50 Image View Detail and type 52 Image View Data records. Similarly, if the x9.37 file contains 4 byte field zero record lengths, then the length prefix of the type 52 record will be updated as well. All of these updated are both applied and reverted as a single modification.

Swap does not save the current file, but instead allows you to view the results to determine if they meet your requirements. You can simply / file / close / after a swap operation to drop the changes that were applied. You should always closely examine your results before saving them to an output file, and you should never overwrite the original file that was input to swap.

Multiple swap operations can be applied to the same file, which is needed when there are numerous images to be swapped that do not exist within a single consecutive range. We would suggest that you save to a work file between each of these modifications.

Import

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

Import allows a new file to be created from a user supplied CSV file which represents the exact content of the new file on a record by record and field by field basis. The input CSV contains one for each data record, with each record field then listed within each row. For example,

The leading records on an X9 file would appear as follows:

```
01,03,T,123456780,123456780,20141017,1201,N,VIEW,VIEW,A,,,
10,01,123456780,123456780,20141016,20141017,1201,I,G,1,X9ASSIST,,C,,,
20,01,123456780,123456780,20141016,20141017,57000000,1,,123456780,,
25,,,05777093,0,20915353/7837,0000010000,44000000,G,8,1,Y,03,0,B
```

Similarly, the leading records on an ACH file would appear as follows:

```
1,01,555555550,123456780,180402,0951,A,094,10,1,Test Bank,TestCo,SD1300
5,200,TestCo,Discretionary,TestCo1,PPD,TestDesc1,,180402,,1,55555555,0000001
6,22,12345678,0,12345,0000010011,id1,name1,d1,1,123456780000001
7,05,transaction1,0001,0000001
```

Import requires that you provide the exact values for each field within each record. Import takes your field values and applies construction rules to ensure that individual fields are populated within each record appropriately. For example, if a field is required as numeric zero filled, then the value will be padded with zeroes on the left as required by the data specification. Another example is justification, where fields are required to be justified left or right and then blank filled. The Import process performs the appropriate justification and padding per all field specifications.

Although you must provide every record in every field, you do not need to provide the accumulated values for the summary counts and amounts in the various trailer records. A value of zero can be supplied instead as proxy for those various fields, and then indicate that auto-repair of the trailer records should be run as part of the import process. When you set the trailer counts and amounts to zero, you must then select the trailer repair option on the import panel. Note that this repair will not run all available repair actions, but will ONLY repair the summary accounts and amounts in the various file trailer records.

Importing a Previously Exported File

Import can be used to create a new file from a file that was previously exported, since they share the same CSV format. This is a very common and logical approach to create a new file when you need to apply complex or repetitive data modifications. In this scenario, you can export a file, modify the created CSV file using various data manipulation tools, and then import the resulting CSV file to create a new and modified result.

CSV Format

Some tools that create CSV files will include column definitions as header rows on the front of the created file. If you have such a file and cannot easily remove those header lines, import will allow you to provide the number of lines to be skipped. Remember that the first line being imported must always be a file header record.

CSV files will typically have all fields enclosed in single or double quotes, with fields separated by either commas or tab characters. Import supports these various formats and will automatically identify and accept those characters appropriately when present.

Check Images

The x9 output file created by the Import will always contain images. Images can either be imported from external files, or they can be dynamically inserted by X9Assist. If you provide the images, then a separate tiff file must exist for each front image and back image.

There are several available options. You can directly provide the actual images that you want to include in the x9 file. A separate front image and back image must be provided for each check. Your images will be included and inserted into the x9 file being generated. The import operation will be aborted if your image cannot be found, or if it is determined that the image you have provided is not a valid tiff format and hence is not usable.

You can alternatively insert replacement images for each front and back image using one of the following options:

- Insert missing image documents: missing image documents will be inserted. These images can be customized with your specific missing image design.
- Insert missing image documents that include MICR lines: missing image documents will be inserted, where the front image will contain MICR lines that are constructed from the x9 data.
- Dynamically draw the images from the imported x9 data: images will be redrawn using the information that is being assigned to the type 25 or type 31 check detail record. X9Assist will ensure that the image meets tiff standards. You can select the check format (artwork) on the image templates tab. You can also provide additional customization using the “draft check” tab within program options.

Record Type 52

The x9 check image record is record type 52 (image view data record).

Field 52.18 (check image length): your value for this field is ignored; Import will assign this length automatically from the image itself.

Field 52.19 (check image): there are several options for this field:

- The images can be located within a single folder. In that case, the image file name can be unqualified, and you can instead select the folder which contains the images. Each check will have two images (front and back) and hence they must be uniquely named.
- Images can instead be located in multiple folders. In this case, you would not select an import folder, but would instead provide a fully qualified image file name for each check image.
- You should use a file naming scheme that defines the check number and the logical image side. An example would be to name your image files “Check_nnnnn_sssss.tif”, where nnnnn is the check number or record number within the overall x9 file; sssss is either “front” or “back”.

Import Data Example

This import example is included in the examples folder and includes the check images. You can actually import the file and examine the results.

```
01,03,"T",555555550,555555550,20130331,0537,"N","TEST","TEST","A",,,1
10,01,555555550,555555550,20130329,20130331,0537,"I","G",1,"X9Assist",,"C",,,
20,01,555555550,555555550,20130329,20130331,5300000000,1,,555555550,,
61,1234,,555555550,"123456789/",0000050020,412564320,"G",,3,,,
50,1,555555550,20130328,00,00,0000000,0,00,0,,,,0,,,,
52,555555550,20130329,,44000000,,,,0,,,,,0000,,0,,0001502,"Bundle_000003/Image_000006_amount_50020_front.tif"
50,1,555555550,20130328,00,00,0000000,1,00,0,,,,,0,,,,
52,555555550,20130329,,44000000,,,,0,,,,,0000,,0,,0002153,"Bundle_000003/Image_000008_amount_50020_back.tif"
25,,,12777012,9,"16808908/9084",0000010000,44000000,"G",0,1,"Y",03,0,"A"
26,1,047770801,20130328,44000000,,,,"Y",0,,,
28,01,097770330,20130328,1,"N",0,0,"A",,,
28,02,127770129,20130328,2,"N",0,0,"A",,,
50,1,555555550,20130328,00,00,0000000,0,00,0,,,,,0,,,,
52,555555550,20130329,,44000000,,,,0,,,,,0000,,0,,0006161,"Bundle_000003/Image_000014_amount_10000_front.tif"
50,1,555555550,20130328,00,00,0000000,1,00,0,,,,,0,,,,
52,555555550,20130329,,44000000,,,,0,,,,,0000,,0,,0001816,"Bundle_000003/Image_000016_amount_10000_back.tif"
25,,,07777098,1,"47872929/4691",0000010002,44000001,"G",1,1,"Y",03,1,"B"
26,1,117770236,20130328,44000001,,,,"Y",1,,,
28,01,027770711,20130328,3,"N",1,1,"B",,,
28,02,037770934,20130328,4,"N",1,1,"B",,,
50,1,555555550,20130328,00,00,0000000,0,00,0,,,,,0,,,,
52,555555550,20130329,,44000001,,,,0,,,,,0000,,0,,0006109,"Bundle_000003/Image_000022_amount_10002_front.tif"
50,1,555555550,20130328,00,00,0000000,1,00,0,,,,,0,,,,
52,555555550,20130329,,44000001,,,,0,,,,,0000,,0,,0001816,"Bundle_000003/Image_000024_amount_10002_back.tif"
25,,,02777097,3,"41344424/7958",0000010004,44000002,"G",2,1,"Y",03,2,"C"
26,1,037770565,20130328,44000002,,,,"Y",2,,,
28,01,127770572,20130328,5,"N",2,2,"C",,,
28,02,057770846,20130328,6,"N",2,2,"C",,,
50,1,555555550,20130328,00,00,0000000,0,00,0,,,,,0,,,,
52,555555550,20130329,,44000002,,,,0,,,,,0000,,0,,0006096,"Bundle_000003/Image_000030_amount_10004_front.tif"
50,1,555555550,20130328,00,00,0000000,1,00,0,,,,,0,,,,
52,555555550,20130329,,44000002,,,,0,,,,,0000,,0,,0001816,"Bundle_000003/Image_000032_amount_10004_back.tif"
```

```
25,,,10777085,1,"58149612/6870",0000010006,44000003,"G",3,1,"Y",03,3,"D"
26,1,127770242,20130328,44000003,,,,"Y",3,,,
28,01,127770187,20130328,7,"N",3,3,"D",,,
28,02,027770892,20130328,8,"N",3,3,"D",,,
50,1,555555550,20130328,00,00,0000000,0,00,0,,,,0,,,,
52,555555550,20130329,,44000003,,,,0,,,,,0000,,0,,0006127,"Bundle_000003/Image_000038_amount_10006_front.tif"
50,1,555555550,20130328,00,00,0000000,1,00,0,,,,0,,,,
52,555555550,20130329,,44000003,,,,0,,,,,0000,,0,,0001816,"Bundle_000003/Image_000040_amount_10006_back.tif"
25,,,03777062,0,"44594284/3463",0000010008,44000004,"G",4,1,"Y",03,0,"E"
26,1,117770388,20130328,44000004,,,,"Y",4,,,
28,01,127770899,20130328,9,"N",4,0,"E",,,
28,02,027770591,20130328,10,"N",4,0,"E",,,
50,1,555555550,20130328,00,00,0000000,0,00,0,,,,0,,,,
52,555555550,20130329,,44000004,,,,0,,,,,0000,,0,,0006138,"Bundle_000003/Image_000046_amount_10008_front.tif"
50,1,555555550,20130328,00,00,0000000,1,00,0,,,,0,,,,
52,555555550,20130329,,44000004,,,,0,,,,,0000,,0,,0001816,"Bundle_000003/Image_000048_amount_10008_back.tif"
70,0005,000000050020,000000050020,00012,,
90,000001,00000005,00000000050020,000000012,"File Generator",20130331,
99,000001,00000051,00000005,0000000000050020,,,
```

Export

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

Export allows you to export a variety of data types to external CSV, text, and xml files. Exported information can include data and/or images. These exported files can be used for more extensive data analysis or for input to other applications. Export supports a variety of output formats including an exact record-field version of the data (which can be used as Import input); a structured field format which is parsed into all commonly needed data fields; and an export into xml that can be used as is or externally translated to other xml schema formats. Additionally, exports can be done from a single file or from multiple files. When running in multi-file mode, export can be used to extract data from a large number of files into a single output file.

Tabs within the export panel allow you to select the type of export being performed along with parameters which further define and control the export process. Export can automatically limit the scope of the exported data in several ways:

- You can specify a range of records to be applied to the export by entering the low and high record numbers. This will limit the export to a certain part of the file (for example, just one bundle or a group of items).
- You can specify that the export should be limited to certain record types (for example, only check detail records).
- You can combine these two selection capabilities (for example, only export the check detail records within a single bundle).

The input filename can be optionally inserted into the first column of the CSV export file. This option is helpful in several situations. First when you are exporting information from various files over a period of time and you need a mechanism to remember and record the origination source. Second is when you are exporting data from multiple files in a single export run, and similarly need to identify the file source for each exported row.

Single Versus Multiple File Exports

Most typically, a single file is being exported. In this case, the file that is exported will be the file that is currently loaded into the viewer, hence there is no need for a file selection dialog.

However, the export facility also allows you to select multiple input files to be exported in a single operation. This is a powerful function since it can read data from a large number of files to create your desired results. These capabilities should be used cautiously, since these exports can run for an extended period of time. This is especially true when there are a large number of files that are selected, when the files themselves are very large, or when the input files exist on external servers and must be read across a network.

Quoted Fields

Your CSV file can be created on either a “standard” or “quoted” basis.

- Our standard CSV format will only enclose a field within quote marks when it contains non-numeric data. Additionally, fields that are quoted will be trimmed to remove leading and trailing spaces. The result is that only the actual data remains.
- Our fully quoted CSV format will write all fields in quoted form and will preserve the leading and trailing spaces within each field. This format can be used if you need your CSV file to contain all data from each field and not be truncated in any fashion.

CSV Tools

You can use exported files as input to other applications such as Excel or your proprietary application systems. Export is an excellent tool to allow you to create CSV files that are shareable and can be used in a large variety of ways.

However, although Excel is a great spreadsheet tool, it does not fare well when used against CSV input. Because of that, X9Ware has developed our own Csv Editor, which is part of our X9Validator (AchValidator) and X9Assist (AchAssist) products. Alternative tools are Libre Office, NotePad, and NotePad++. These tools include search facilities allowing you to find data on a string basis, which can be very helpful when doing research.

Export Layouts

This table represents the available export formats with their attributes and common usage.

	Description	Images Included?	Importable?	Selected Record Types?	Comments
(1)	CSV in native record format.	No	Yes, but only if the images are drawn as proxies since the actual images are not exported along with the data, and then also only when all record types are exported.	Yes	This format contains the field values within the selected record types per the file that is currently loaded and the associated rules specification. It provides a full representation of the data content for the selected record types, and a complete data representation of the entire file when all record types are selected. A common usage is to limit the export to certain record types based on need. For example, the export might be limited to the type 01 file header, the type 25 check records, and the type 26-28 addenda. The output line

	Description	Images Included?	Importable?	Selected Record Types?	Comments
					numbers will match back to the input file when all record types have been selected.
(2)	CSV in native record format with images.	Yes	Yes, but only when all record types have been exported.	Yes	This format contains the field values within the selected record types per the associated rules standard. The output line numbers will match back to the input file when all record types have been selected. This export file can be modified using various standard editor tools and then imported to create a new file which has been changed per user specific requirements. For example, the order of certain record types within the file can be changed; records can be removed; individual fields can be modified. Repair can also be run after the file is imported to correct trailer records when desired.
(3)	CSV parsed items into fixed fields.	No	No	Yes	Items are parsed into individual fields and exported into a fixed column format. This format can be easier to process since the items are populated on a standard basis into fixed columns. This item format is constant and does not vary regardless of the file type (forward presentment or returns) and the associated rules standard (x9.37 versus x9.100-180). Data fields for non-item groups (file headers, cash letter headers, bundle headers, etc) can be optionally included and will be presented in their native format.
(4)	CSV parsed items into fixed fields with images.	Yes	No	Yes	Items are parsed into individual fields and exported into a fixed column format.
(5)	CSV parsed items into	Yes	No	No	Items are parsed into individual fields and exported into a fixed column

	Description	Images Included?	Importable?	Selected Record Types?	Comments
	fixed fields with images, with a descriptive column heading row inserted as row one.				format. The first CSV row will contain column names. Only item level record types (those attached to debits and credits) will be exported; header and trailer records will not be exported. This is because column content must be in true fixed sequence to match the row zero column names.
(6)	CSV record groups into variable columns.	No	No	Yes	Output is constructed on a record group basis, where each group is the owner record type with all attached subordinate types. This format can be easier to parse since the output records for each group are concatenated into a single string which eliminates the need to separate the rows into record groups in your parser. This format has the further benefit that all output fields are present. Note that record types can be excluded and will not appear within the output. For example, you can exclude record type 50-52 if that data is not needed.
(7)	Tiff tag information.	No	No	No	This export format provides a vision into the tiff tags that are present for each image and is useful when performing a detailed image analysis for a specific originator or capture system.
(8)	Record data.	No	No	Yes	Records are exported in their text (txt) form, which provides access to their record data (eg, typically 80 bytes long) in the exact format as present on the input file. When using this export format, there is an option to append the record number as either a suffix or prefix to the written data. Appending the record number allows you to sort the data on various fields and still be able to resort the records

	Description	Images Included?	Importable?	Selected Record Types?	Comments
					into their original order. The record number also allows you to trace every record back to the original location within the original file.
(9)	XML.	Yes, in Base64 encoded format	No	Yes	Export to XML builds an output xml file that can be processed by other xml enabled applications. Please advise if you have requirements for this functionality and we would be glad to work with you on building new xml formats for specific vendor applications.
(10)	Errors.	No	N/A	N/A	Errors for the currently loaded file are exported and shared with others. All selection criteria applies (record number range and record types). The export can be limited by type (data or image) or severity (error, warn, and info). Output is in a fixed format which can be easily analyzed.
(11)	Use cases.	No	N/A	N/A	Use cases which can be used by Make and Scrub when creating test data. This export facility can be used in several manners. First is to create a use case file from a single file. Second is to merge the use case content from the current file into an aggregated use case file that is being constructed using data from a series of files. When that is done, the additional files are added one at a time. Duplicate use case entries will be removed and only new use case entries will be added to the aggregated use case file.

Type 52 Images (Applies to X9)

Export will insert the name of the each image file into its corresponding type 52 image view data record. The name is stored into field 5.19, which is normally houses the image data. By putting the check image file name into field 52.19, Export is able to logically associate each image with its

type 52 record. Note that when using the x9.100-180 standard, images are stored in field 52.27, which is fully supported by this process.

When the type 52 record is exported, the image field is updated to contain one of the following values, subject to the selected options:

- Absolute file name, which contains the drive and path (folder) of each image file along with the actual file name. It is recommended that you use absolute file names since they fully describe the output file location. Absolute file names are required when the exported file is used by certain internal functions such as Generate.
- Relative file name, which contains the base file name only (it does not include the path). Relative names are useful when the exported CSV and images will be transported to other environments or systems.
- Base64-basic encodes the image using base64 and inserts the resulting string into the image field. This eliminates the external storage of the image in the file system and can simplify access to the image data. Base64 image exports run substantially faster than exports into the file system, since the operating system overhead to update the file system is eliminated. Image size is typically increased by 30-40 percent when using this encoding format.
- Base64-mime is similar to base64-basic encoding, just based on the MIME format.

Exporting Images (X9 Files Only)

When exporting images, you will be directed to select your output image folder. Export will then create a new sub-folder for each file within this high level folder. This folder creation applies for the export of just one file, to make the process consist with multi-file export.

Additional sub-folders are then created for each bundle using the record number of the type 70 bundle record. This has two advantages. First that the created bundle folder name will be unique and can be directly associated with the input file. Second that it allows the bundle folders to remain in ascending sequence for the overall file.

Exported Image Formats (X9 Files Only)

When exporting images, you can optionally select the image format that will be created, which can be TIF, PNG, JPG, or GIF. It is important to realize that embedded images within x9.37 files are in TIF format. This means that when images are written in TIF format (which is the default), they can be written exactly as they are contained within the x9.37 file and do not need to be converted in any manner. However, export also allows you to select an alternative format. When that is done, images must be converted from the TIF format to the user selected format. Please realize that image conversions can be time consuming, with PNG probably being the better alternative given its performance and compression. PNG and JPG images will retain their original size and DPI. Since GIF images to not have an internally defined DPI, they will be standardized to 200 DPI as a matter of convenience. Image conversions should be utilized carefully and only when absolutely needed. An alternative is to export images in TIF format and then subsequently utilize a batch conversion strategy to convert them later, perhaps when importing them into an archive or user application.

Output Image Folders (Applies to X9)

A high level output folder is provided to the export images function. A new sub-folder will be created within the selected folder using the current file, which will then own a series of bundle level folders. Each bundle folder is named using the record number for that specific bundle, which is unique and preserves the relative position of the bundle within the file. This common file level approach supports either single or multi-file export.

Export includes an option to automatically clear an existing image folder, and it is highly recommended that this default is selected. If an output folder exists which is not cleared, then new bundles and image files will be created and merged into that folder structure which may overwrite existing files and can result in a confusing series of files and folders.

Export as Items into Fixed Columns (Applies to X9)

Items can be parsed into logical field content and then exported into fixed columns. This format can be easier to parse since the location of individual data columns will be fixed which can simplify your subsequent parsing of this data. Specifically, the type 25 and type 31 records are parsed into logical items that will contain the following columns:

Column Number	Fixed Column Data Content for the Current Item
1	Record type
2	Amount
3	Item sequence number
4	Routing
5	MICR OnUs
6	MICR Auxiliary OnUs
7	MICR Epc
8	Documentation type indicator
8	Returns Acceptance Indicator
10	MICR valid indicator
11	BOFD indicator
12	Addendum count
13	Correction indicator
14	Archive type indicator
15	Credit account
16	Return reason
17	Forward bundle date

18	Return notification indicator
19	Payor bank name
20	Payor bank business date
21	Payor account name
22	Field4 parsed from the item MICR OnUs field
23	Account parsed from the item MICR OnUs field
24	Process Control parsed from the item MICR OnUs field
25	Reserved-1
26	Reserved-2
27	Reserved-3
28	Image creator date from the front image
29	Image creator routing from the front image
30	Image reference key from the front image
31	Front image name (when exported)
32	Back image name (when exported)
33-45	First primary (26 or 32) or secondary (28 or 35) endorsement
33	Record type
34	Routing
35	Endorsement date
36	Item sequence number
37	Deposit account number
38	Deposit branch
39	Payee name
40	Truncation indicator
41	Conversion indicator
42	Return reason
43	Endorsing bank identifier
44	User field
46-58	Second endorsement
59-71	Third endorsement
72-84	Fourth endorsement
Etc	Continued for as many endorsements that exist for this item

Export as Native Record Format

Native format follows the current rules specification. For example, an input file encoded per the x9.37 DSTU specification will be exported per the record layouts that are defined by that standard. You must reference the associated file specification to obtain a list of the fields that are exported by record type. These fields will be the same as displayed within our file viewer. This export format has the advantage that it covers all record types and all fields.

When using this export format, there is an additional option to append the record number as either a suffix or prefix to the written data. Appending the record number allows you to sort the data on various fields and still be able to resort the records into their original order. The record number also allows you to trace every record back to the original location within the original file.

Export as Groups into Variable Columns

Records can be exported as groups and not individual record types. A record group consists of the owner record type (for example, a type 25 or type 31) followed by all records that are attached to that owner. Although the concept of record groups only applies to items, it also logically applies to other record types when they are the owner of type 68 user records. For example, a record group might consist of only a single type 01 file header, but it could also consist of a type 01 file header followed by two type 68 user records.

Record groups are exported as a CSV of all record types and fields within the group. For an item, this single CSV row might contain fields for the following record types: 25, 26, 28, 52, 54, 52, and 54. Also remember that record types can be excluded from this process. For example, you can exclude record types 50 and 52 from the export, and in that case this same record group would be exported as: 25, 26, 28, and 28. If you only need the item with the attached BOFD endorsement, then you can exclude 28, 50 and 52, and then the exported record group will consist of just the 25 and 26 records. Although these are examples of forward presentment files, the process applies equally to return files.

This export format is provided as a convenience since it may be easier to parse by your application programs. Although the concept of record groups is shared with item export, it has the benefit that it includes every field with the individual record types will be exported.

Export As Errors

Errors can be exported in CSV format to allow you to get a list of errors associated within a given file that you can easily share with others. All of the previously stated selection criteria applies (record number range and record types).

In addition, error export allows you to indicate that you want to limit the export based on:

- All errors
- Record (data) errors only
- Image (tiff) errors only

You can also identify the severity of the errors to be exported. This can be all error severities, or combinations of error level, warn level, and info level. The field data that is written is aligned per the columns as depicted on the Errors tab. This list of fields is as follows:

1) Error description
2) Error identifier
3) Error record number
4) Error field number
5) Error field name
6) Error field data position
7) Error field data length
8) Error field value
9) Error field mandatory indicator
10) Error field list of allowable values
11) Error field primary edit rule
12) Supplemental information
13) Item amount
14) Item ECE sequence number
15) Item routing number
16) Cash letter record number
17) Cash letter identifier
18) Cash letter amount
19) Cash letter business date
20) Bundle record number
21) Bundle amount
22) BOFD routing number
23) BOFD business date
24) BOFD sequence number
25) Formulated error message which best summarizes the error condition

Export as Use Cases

Use case files are input into various X9Assist functions, and very specifically Scrub and Make. In particular, the basic use case files which are utilized by Scrub contains four fields:

- Routing
- Account number
- Process Control
- Auxiliary OnUs

Export Use Cases creates a file that is compatible with the above and contains the following fields:

- Routing
- Account number
- Process Control
- Auxiliary OnUs
- Amount
- Item sequence number

- EPC
- Documentation type indicator
- Return acceptance indicator
- MICR valid indicator
- BOFD indicator
- Correction indicator
- Archive indicator
- Return reason
- Return notification indicator
- BOFD routing

The export use cases function allows you to create these lists using your production data, with data extracted from one or more files. The creation of these use case files from your production files will guarantee that valid MICR line combinations are used by X9Assist when creating sanitized checks. This will help to ensure that your downstream application systems will be able to process the sanitized data and will not arbitrarily reject items based on MICR content.

This function allows you to either create a new use case file or to merge data into an existing file that you have previously created. By merging into an existing use case file, you can build use case files which contain a large number of routing and account numbers. The merge process ensures that any given routing/account number combination will only appear once in the accumulated use case file, no matter how many times it may appear in the individual files that you are using for this extract and accumulation process.

Using this function, you may want to create use case files (for Scrub) that could be used for a variety of different functions. For example, you may want to have different use case files for forward presentment versus returned items files. You can then use these accumulated files for common functions such as Scrub, as an alternative to the vanilla files that can instead by created using the Use Case Editor.

Export as Tiff Tags (Applies to X9)

The IFD directories from Tiff images can be exported into CSV format. The export is in CSV format and will contain one row for each tff tag within each image. The content of each exported row is as follows:

- Record number
- Tiff tag index
- Offset of the tiff tag within the image
- Tag
- Type
- Count
- Value
- Directory entry in hex

ACH Standard Entry Classes (SEC)

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**NO**	**YES**	**NO**

AchValidator/AchAssist supports all ACH Standard Entry Classes. This includes the ability to parse and display these transactions on a record and field basis, as well as to apply all validations that are required for this data.

A Standard Entry Class Code (SEC Code) identifies a specific payment application. The code incorporates type of payment (debit or credit), account nature (consumer or corporate), and record format used to carry format. Implicit in the code is how authorization is arranged (standing, oral, type of transaction, etc). The list can be broken into three parts: Consumer classes, Corporate classes, and classes that are applicable for both types of accounts.

Consumer transactions are most commonly credits (payroll, retirement, dividend, interest payments, etc). There are also debit applications as well (insurance premiums, mortgage payments, utility payments, etc). Transactions can be a single payment event or recurring.

Credits (with the exception of reversals) are not permitted in standard entry class codes: ARC, BOC, POP, RCK, TEL, WEB, and XCK.

SEC	Description	Usage
ACK / ATX	Acknowledgement Entries	These optional Standard Entry Class Codes are available for use by the RDFI to acknowledge the receipt of ACH credit payments originated using the CCD or CTX formats. These acknowledgments indicate to the Originator that the payment was received and that the RDFI will attempt to post the payment to the Receiver's account. Acknowledgment entries initiated in response to a CCD credit entry utilize the ACK format. Acknowledgments initiated in response to a CTX credit entry utilize the ATX format.
ADV	Automated Accounting Advice	Represents an optional service to be provided by ACH Operators that identifies automated accounting advices of ACH accounting information in machine readable format to facilitate the automation of accounting information for Participating DFIs. The standard entry class for advices must be 225 and the transaction code must be 24 (which represents a zero dollar item).
ARC	Accounts Receivable Entry	This Standard Entry Class Code enables Originators to convert to a SingleEntry ACH debit an eligible check received via the U.S. mail or at a drop box location for the payment of goods or services. The Receiver's source document (i.e., the check) is used to collect the Receiver's routing number, account number, check serial number, and dollar amount for the transaction. Authorization for an ARC entry is obtained through notice provided to the Receiver by the

SEC	Description	Usage
		payee and the Receiver's going forward with the transaction.
BOC	Back Office Conversion Entry	This Standard Entry Class Code enables Originators, during back office processing, to convert to a SingleEntry ACH debit an eligible check received at the point of purchase or manned bill payment location for the in person purchase of goods or services. The Receiver's source document (the check) is used to collect the Receiver's routing number, account number, check serial number, and dollar amount for the transaction. Authorization for a BOC entry is obtained through notice provided by the Originator at the point of purchase or manned bill payment location and the Receiver's going forward with the transaction.
CCD	Corporate Credit or Debit	This application can be either a credit or debit application where funds are transferred between unrelated corporate entities, or transmitted as intra-company cash concentration and disbursement transactions. This application can serve as a standalone funds transfer, or it can support a limited amount of payment related data with the funds transfer.
CIE	Customer Initiated Entry	Customer Initiated Entries are limited to credit applications where the consumer initiates the transfer of funds, typically to a company for payment of funds owed to that company, through some type of home banking product or bill payment service provider.
COR	Automated Notification of Change or Refused Automated Notification of Change	This Standard Entry Class Code is used by an RDFI or ODFI when originating a Notification of Change or Refused Notification of Change in automated format. It is also used by the ACH Operator that converts paper Notifications of Change to automated format.
CTX	Corporate Trade Exchange	The Corporate Trade Exchange application supports the transfer of funds (debit or credit) within a trading partner relationship in which a full ANSI ASC X12 message or payment related UN/EDIFACT information is sent with the funds transfer. The ANSI ASC X12 message or payment related UN/EDIFACT information is placed in multiple addenda records. CTX entries may be accompanied by multiple addenda records that may include information such as invoice numbers or other payment related information.
DNE	Death Notification Entry	This application is utilized by a Federal Government agency (e.g., Social Security Administration) to notify a depository financial institution that the recipient of a government benefit payment has died.
ENR	Automated Enrollment Entry	ACH enrollment entries originated by a DFI at the request of an account holder at the DFI. Entries are sent to a Federal Government Agency. (See Payment Related Information in this Appendix Two.) These entries can be used for Social Security enrollment.

SEC	Description	Usage
IAT	International ACH Transaction	This Standard Entry Class Code identifies an ACH credit or debit entry that is part of a payment transaction that involves a financial agency's office that is not located within the territorial jurisdiction of the United States. These international payments convey specific information defined within the Bank Secrecy Act's "Travel Rule" to ensure that all parties to the transaction have the information necessary to comply with U.S. law, which includes the programs administered by the Office of Foreign Assets Control (OFAC).
MTE	Machine Transfer Entry	The ACH Network supports the clearing of transactions from Automated Teller Machines. These credit or debit entries initiated at an electronic terminal, as defined in Regulation E, to effect a transfer of funds to or from a deposit account of an Originator; for example, an ATM cash deposit or withdrawal. NOTE: Credit entries so initiated to the accounts of third parties are CIE entries and are to be formatted as such.
POS	Point of Sale Entry	Debit entries initiated at an electronic terminal as defined in Regulation E of The Board of Governors of the Federal Reserve System to pay an obligation incurred in a point-of-sale transaction, or to effect a transfer of funds from a deposit account (for example, a point-of-sale terminal cash withdrawal), and reversing, adjusting, and other credit entries relating to such debit entries, transfer of funds or obligations. POS entries are originated in a non-shared system in which no agreement other than these Rules exists between the ODFI and the RDFI, and in which transactions are typically initiated by use of a merchant issued plastic card.
PPD	Pre-arranged Payment and Deposit Entry	Direct Deposit or Pre-authorized Bill Payment. Direct deposit is a credit application that transfers funds into a consumer's account at the Receiving Depository Financial Institution. The funds being deposited can represent a variety of products, such as payroll, interest, pension, dividends, etc. Pre-authorized payment is a debit application. Companies with billing operations may participate in the ACH through the electronic transfer (direct debit) of bill payment entries. Through standing authorizations, the consumer grants the company authority to initiate periodic charges to his account as bills become due. This concept has met with appreciable success in situations where the recurring bills are regular and do not vary in amount insurance premiums, mortgage payments, and installment loan payments being the most prominent examples. Standing authorizations have also been successful for bills where the amount does vary, such as utility payments.

SEC	Description	Usage
POP	Point-of-Purchase Entry	A Single Entry debit used by Originators as a method of payment for the in person purchase of goods or services by Receivers. These SingleEntry debit entries are initiated by the Originator based on a written authorization between the Originator and Receiver and notice provided by the Originator at the point of purchase or manned bill payment location. The source document, which is voided by the merchant and returned to the Receiver at the point of purchase, is used to collect the Receiver's routing number, account number, and check serial number that will be used to generate the debit entry to the Receiver's account. This type of entry may only be used for non-recurring, in-person (ie, at the point-of-purchase) entries for which there is no standing authorization with the merchant for the origination of ACH entries to the Receiver's account.
RCK	Re-presented Check Entry	A represented check entry is a SingleEntry ACH debit application used by Originators to represent a check that has been processed through the check collection system and returned because of insufficient or uncollected funds. This method of collection via the ACH Network, compared to the check collection process, provides Originators with the potential for improvements to processing efficiency (such as control over timing of the initiation of the debit entry) and decreased costs.
SHR	Shared Network Transaction	Debit entries initiated at an electronic terminal as defined in Regulation E of The Board of Governors of the Federal Reserve System to pay an obligation incurred in a point-of-sale transaction, or to effect a transfer of funds from a deposit account (for example, a point-of-sale terminal cash withdrawal), and reversing, djusting, and other credit entries relating to such debit entries, transfer of funds or obligations. SHR entries are originated in a shared system where an agreement, in addition to these Rules, exists between the ODFI and RDFI, and in which the transactions are typically initiated by the use of a plastic card issued by the Receiver's DFI.
TEL	Telephone-Initiated Entry	A Single-Entry debit initiated by an Originator pursuant to an oral authorization obtained over the telephone to effect a transfer of funds from a Consumer Account of the Receiver. This type of entry may only be used for a Single Entry for hich there is no standing authorization for the origination of ACH entries to the Receiver's account. A TEL entry may only be used when there is an Existing Relationship between the originator and the Receiver, or, when there is not an Existing Relationship between the Originator and the Receiver, when the Receiver initiates the telephone call.
TRC / TRX	Truncated Entries	Transactions for truncated checks being safe kept by the keeper bank (Originator) as defined by a check truncation program. The TRX format allows financial institutions to use a single entry to carry information from multiple checks.

SEC	Description	Usage
WEB	Internet-Initiated Entry	Debit entries initiated by an Originator pursuant to an authorization that is obtained from the Receiver via the Internet to effect a transfer of funds from a Consumer Account of the Receiver. This application class helps to address unique risk issues inherent to the Internet payment environment through requirements for added security procedures and obligations.
XCK	Destroyed Check Entry	Debit entries initiated in the event an item eligible for Article Two, section 2.7 (Destroyed Check Entries) is contained within a cash letter that is lost, destroyed, or otherwise unavailable to and cannot be obtained by the ODFI.

X9Utilities Console

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**NO**	**NO**

The X9Utilities Console is an interactive tool that is available within both X9Utilities and X9Assist. The console is a powerful facility that allows command line parameters to be entered, validated, and then submitted for execution. It serves as a good demonstration of the overall capabilities of X9Utilities, since it allows the command line to be constructed visually, with validation of files and command line switches as they are entered. The console allow X9Utilities to be run on demand. It intercepts and displays the system log as x9utilities is executing. It then displays the final exit status that is posted by the x9uti.exe.

The console is an integral part of X9Utilities itself, and launched using the "-console" switch on the command line. By using this approach, it is not a separate program but instead embedded within the standard x9utilities runtime.

The console allows allows a set of parameters can be save to xml as a "run unit" that can be subsequently loaded an executed again. This is a provided as a productivity tool, since it allows a given set of parameters to be saved and executed again in the future, as needed.

As previously mentioned, X9Utilities is a command line tool. In your production environment, it will be invoked on a batch (command line) basis. This might be from a scheduled batch script, or more possibly directly by an application program. The console can serve as a workbench for your testing. It provides an illustration of x9utilities capabilities and operation.

The console is made available as part of X9Assist since that product is a desktop tool, is designed with a UI (User Interface) for human interaction, and is our full-function desktop product. X9Assist customers can use the console to actually run X9Utilities interactively in their environment. An X9Utilities license must be purchased if you subsequently decide that you would like to run the tool on batch basis.

This documentation describes how to use the X9Utilities Console. However, it does not provide detailed information on each of the functions that are available via that batch (command line) utility. Please reference the X9Utilities User Guide (which is available in its entirety via help) for complete information regarding functions, capabilities, and command line options.

X9Utilities Evaluations

The console can be used during X9Utilities evaluations, to both simplify and expedite the review process. Product evaluations are normally performed within a limited time frame and hence productivity factors are important. The console helps greatly during this process, since you can run any X9Utilites function from start to end within minutes, without the need to develop a batch script. You can also repeat the same test using a variety of input files, and can save the work unit

for further repeated testing at a later time. All of these capabilities can add a lot of value during an evaluation of our X9Utilities product, so it represents a great way to get started. Once you have determined that your evaluation is favorable, X9Ware can then provide an evaluation copy of X9Utilities, which will allow you to perform the same operations on a command line basis.

X9Utilities Batch File (Windows)

The console can be invoked using a batch file. A sample of this is included as part of the Windows installation, in folder: / samples / console / .

```
@echo off

: Run the x9utilities using a start command, which allows it to run separately and
the consolse window itself to close.

cd "c:/Program Files/X9Ware LLC/X9Utilities R4.11"
start "" x9util -console

: pause

exit /b
```

Console Fields

The X9Utilities Console contains the following fields which allow you to identify the function to be performed along with their associated parameters:

- “Function” is a drop down box which allows the X9Utilities task to be selected.
- “Batch” is a check box that is used to indicate that batch processing is activate. This is a more complex runtime mode, where a command is executed against all of the files within a folder (instead of against a single file). When using batch mode, the input file must be defined as as wild-card string, where an asterisk is used for pattern matching during the file selection process. Refer to the topic “Batch/Script Operations” in the X9Utilities User Guide for more information.
- “Threads” applies only when running in batch mode, and identifies the number of threads that will be used during work unit processing.
- “Wild Card Pattern” applies only when running in batch mode, and defines the pattern that will be applied when selecting files within folders. An example might be *.x9.
- “Command Switches” is a list of switch values which are to be applied to the selected task. The entered switches must begin with a dash and must be valid for the selected function to be performed. Refer to the X9Utilities User Guide for more information.
- “Rules Specification” is used to select the x9.37 specification to be associated with the selected task. For example, the specification is critical for the “-write” since it identifies the x9 rules to be utilized, which identifies key information such as fields and field alignments.

- "Command line as executed" is a display box which summarizes the command line that will be executed for the selected task. The command line will be updated interactively as changes are made to the above fields (function, switches, rules, etc).

Input and Output Files

The X9Utilities Console includes a facility which must be used to define the input and output files that are processed. There are various files that can be specified, with certain files required based on the function being performed. These files are selected via a file chooser and the fully qualified file name is then displayed in the text box immediately to the right of each selection box. The chosen input and output files will be passed to X9Utilities via the command line. It is also informative to take a look at the command line box to see how input and output files are provided to X9Utilties via the constructed command line.

The following input and output files can be specified:

Function	Input File	Image Folder	Secondary Input File	Output File	Results File
Translate	Mandatory	Optional; will default to "_IMAGES" when not specified		Optional; is assigned based on the input file name when not specified	
Write	Mandatory		Optional; defines the HeaderXml file to be used when not provided as the first row within the input csv file	Optional; is assigned based on the input file name when not specified	
Import	Mandatory	Optional; defines the image folder base name when image file names are relative		Optional; is assigned based on the input file name when not specified	

Function	Input File	Image Folder	Secondary Input File	Output File	Results File
Export	Mandatory	Optional; defines the image folder when image files are written		Optional; is assigned based on the input file name when not specified	
ExportCsv	Mandatory	Optional; defines the image folder when image files are written		Optional; is assigned based on the input file name when not specified	
Validate	Mandatory				Optional; csv of errors that were found; assigned based on the input file name when not specified
Scrub	Mandatory		Mandatory; defines the scrub xml parameters file to be used	Optional; is assigned based on the input file name when not specified	Optional; csv of fields that were scrubbed; assigned based on the input file name when not specified
Update	Mandatory		Mandatory; defines xml file which is used to identify the field level update parameters	Optional; is assigned based on the input file name when not specified	Optional output csv file name
Split	Mandatory		Mandatory; defines xml file which is used to identify the field level split parameters	Not allowed; it is defined within the xml file.	Optional output csv file name

Function	Input File	Image Folder	Secondary Input File	Output File	Results File
Make	Mandatory; defines the input use case file. Along with the input file, you must specify the “-reformatter” and “-generator” switches to define those xml files, which are mandatory as well.		Optional; defines the routing list file when it is not defined within the generator xml file.	Mandatory; defines the output file to be created.	
Merge	Mandatory; defines the input folder that contains the files to be merged.			Mandatory; defines the output file to be created.	
Compare	Mandatory		Mandatory; defines the secondary input file to be compared against		Optional output text file name
ImagePull	Mandatory	Optional; defines the image folder when image file names are relative	Mandatory; defines csv file which is used to identify the item images to be pulled		Optional; defines output csv file which identifies the images that have been pulled.

Work Units

A work unit is a combination of all entered fields which describe the X9Utilities task to be performed. A work unit can be saved to xml and then subsequently loaded at a future time, allowing the work unit to be performed again. The load and save buttons on the right side panel are used for this purpose. Work units are stored in folder / documents / x9_assist / xml / x9utilities /. The ability to save and subsequently reuse work units allows task to be performed repetitively as needed. This might be used to perform a user periodic task and eliminates the need to create a batch script for that same purpose. Work units can not only be stored, but can also organized into subfolders within / documents / x9_assist / xml / x9utilities /.

X9Utilities Console

The console is used to display system log information for each new task that is executed. The console is initialized at the start of each new task and will be updated with log snapshots as the task is run. There are limits as to how much information can be shown within the log, which requires some truncation when the logging becomes excessive. In these situations, logging will be suspended. The console will still contain the very last lines from the log, which is important since these lines may contain output file names, totals, error messages, etc. Console content can be scrolled (using the scroll bar on the right) and can be copied to the clipboard (using CTRL A and then CTRL C). Once copied to the clipboard, the text can be pasted into other external applications as needed for specific purposes.

JDK Requirements

The console requires JDK 11 or higher for execution.

Function Execution

User input is edited interactively and errors and shown when identified. Once a valid work unit has been entered, it can be optionally saved and ultimately executed using the run button on the action line. Many tasks will run very quickly. Depending on the function performed, the number of files involved, and the relative size of those files, the run may take an extended period of time. The console has several indicators to inform you that a utility task is running:

- The background of the X9Utility console is changed from gray to cyan while the task is in a running state, and will be changed back to gray on completion.
- Exit status on the action line will be changed to “running” while the task is in a running state, and will be changed to the numeric exit status on completion.
- The status icon on the action line will be changed to a “runner” image while the task is in a running state, and will be changed back to a green/red icon on completion.

A popup message will displayed on completion that provides the exit status and the name/size of the output file. The elapsed time (in seconds) will also be displayed on the action line.

Command Line as Executed

The console includes a display box which shows the command line as executed. This is provided for several reasons:

- First is that it provides details on how the command line is formulated and thus is very helpful for exposure into the technical workings of the utility. You can compare the content of the formulated command line against the X9Utilities User Guide to fully review the command formats.
- Second is that the command line can be used as the basis for developing batch scripts. This would be done by fully testing the function to be performed, and then by creating a batch

script from the command line. The new batch script may have certain parameters or file names turned into symbolic parameters, increasing both flexibility and usability.

Demonstrations

The following are examples of using the utilities console through a series of sequential operations:

##	Function	Steps needed to run the demonstration
(1)	Export x9.37 data + images into fixed column CSV format.	• You will first need an x9.37 file to be exported. A good example would be to copy our sample file "Test ICL with 10 checks.x9" to your documents folder. You will find this x9 file in the program installation folder; for example in C:\Program Files\X9Ware LLC\ X9Utilities R4.11\samples\files\. The file needs to be copied, because export will (by default) put the output CSV file and image folder next to the input file, so you need to use a folder where you have write privileges. • You will notice a series of file selection rows that are used to identify the input and output files for the current function that is to be executed. These rows select files that are automatically populated into the command line. The first of these rows is for the input file. • Once you have copied your sample x9 file to your documents folder, you then need to use the SELECT button to browse and select this as the input file. • Take a look at the switches; you can change these as desired. You will need to look at the X9Utilities User Guide for more information on the switches that are available. A good first test would be to export into fixed CSV columns with images and create a summary JSON file totals. The switches for this would be entered as: -xf -i -j • Look at the box which contains the command line as it will be executed. This shows exactly how the command line has been constructed based on your input file and entered switch values. • Hit the RUN button in the lower right to execute X9Utilities. • The X9Utilities system log will scroll as the command executes, and will eventually show as "finished" when complete. A popup box will be displayed and you can click OK. • When the export has completed, you will find the exported CSV file and the associated images located in your input folder. • Open the exported CSV file (for example, with a text viewer) and take a look at the export contents. You can use the documentation provided in the X9Utilities User Guide for a definition of the fixed format columns that have been created. • This example has assigned defaults for the image folder (secondary file) and the output file. You can alternatively assign these using SELECT. When you do that, you will also see these file names

##	Function	Steps needed to run the demonstration
		populated into the command line box.
(2)	Export x9.37 data + images into variable column CSV format.	• This is just a variation on the above example (1). The difference is that we will export into a variable number of CSV columns instead of into fixed columns. The variable column format follows the record and field definitions per the x9.37 standards. You will need to refer to the x9.37 standards themselves (eg, the 2003 DSTU) for a definition of this CSV format. • Follow the above example, but change the command line switches to indicate that variable columns should be created: -xc -i -j • Look at the box which contains the command line as it will be executed. This shows exactly how the command line has been constructed based on your input file and entered switch values. • Hit the RUN button in the lower right to execute X9Utilities. • Review output as described in the above export example.
(3)	Import x9.37 from an exported file, or a CSV of your creation.	• This example builds on the above example (2), where the output file is in variable column CSV format. For the purposes of import, this format is mandatory, since import requires that the input file be constructed using the variable (records and fields) layout. This provides a very strict definition of the x9.37 file to be created. • Select the IMPORT function using the drop down box. • Set command line switches to write a JSON summary file and to also include additional logging: -j -l • Use the input file SELECT button to select the variable columns CSV file that was created by example (2) above. • Use the output file SELECT button to identify the output file that is to be written. For example, you can create the output file in the same folder where your input file exists. Following the example, we suggest that you create the output file as “Test ICL with 10 checks imported.x9”. • Review the selected files to ensure that they are correct. • Review the “command line” box for insight into how the import command has been constructed. • Hit the RUN button in the lower right to execute X9Utilities. • Review the created x9.37 file. You will see that it is has the exact same content as the x9.37 file that was originally exported.
(4)	Compare two x9.37 files.	• This example builds on the above examples (2) and (3), where example (2) exported a file to data+images, and example (3) then imported the data+images to create a new x9.37 file. • Given the simplistic flow of those two processes, we would expect the original x9.37 file to be equal to imported x9.37 file, which was constructed from a combined export-import process. • Select the COMPARE function using the drop down box. • Use the input file SELECT button to select the x9.37 that was input to example (2), which was the original file exported into variable column format.

##	Function	Steps needed to run the demonstration
		• Use the secondary file SELECT button to select the x9.37 file that was created by example (3), which was an import of the CSV file in variable column format. • Hit the RUN button in the lower right to execute X9Utilities. • Review the created "_output.txt" and "_output.csv" files, which will contain all identified differences. • Also take a look at the exit status that was posted by X9Utilities, which should be zero.

Reusing a Saved Work Unit

The console can be used to save a work unit to xml for subsequent reuse. This work unit can be used to the batch version of X9Utilities to rerun this command in a batch environment. This can be useful when a function needs to be performed in a batch environment and can be repeated exactly as originally performed via the console. In some situations, this may eliminate the need to develop and test a batch script to perform the same function. The work unit is provided to X9Utilities using the **-workUnit:"-fully qualified file name"** command line parameter. No other parameters or switches are required. You can review the created xml file for more insight into the work unit xml definition and embedded content.

E13B OCR Results Viewer

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**NO**	**NO**

The E13B OCR Results Viewer provides insight into the results of our character recognition process, as applied to the current front-side image within the Item Viewer. This includes the front image itself, image as deskewed, the isolated MICR line, recognition results, and all logging information that was generated by our recognizer. If recognition for a specific image is inaccurate and if you believe that the results should be improved upon, then we would very much like you to use the zip process (below) to create documentation that you can provide to us, which is the very detailed information that we will need to improve the OCR recognition engine. We have implemented this exporting process since it will provide just a single image to us for this analysis (we do not need the entire file). We realize that this will one image will contain PII (personally identifiable information), which you must take into consideration. We will use the data only for analysis purposes. We will then delete it and inform you when it has been removed from our system. This feedback is what we need to provide continuous improvements to our E13B recognition engine.

This OCR technology is proprietary to X9Ware LLC and based solely on our development efforts. The implementation is 100% Java and represents the only E13B OCR solution (that we are aware of) that runs within the JVM. This character recognition is not dependent on JNI (Java Native Interface) wrappers used to execute external machine dependent code (most typically written in C or C++).

Our recognition engine analyzes the image through a series of steps which include skew detection, re-orientation, noise removal, and character matching. The recognition process uses advanced character framing that attempts to match a character despite pixel noise that might surround it, including handwriting that spills into the MICR line area. The matching process is based on both character size and shape. There are both primary and secondary matching algorithms, where the secondary is applied as confirmation only when the primary method is not highly confident.

Code line recognition accuracy approaches 99.8% accuracy when presented with with good quality scanned images. Throughput has been measured at 15 items per second when single threaded and over 50 items per second when running on four concurrent threads (which is the implementation used here within our Item Viewer).

Panel Content

The following images are included within this viewer:

- Original – the original front-side image as routed into the recognition engine.
- Bottom – the bottom portion of the front-side image.

- Deskewed – the bottom of the image in a deskewed formthat has been redrawn using the detected angle.
- Micr Band – the micr band as extracted from the bottom of the deskewed image.
- Micr Area -the isolated micr area as extracted from the micr band.
- Cleaned & Smoothed – the micr area which has gone through cleaning and edge smoothing.
- With Recognition Strip – the original front-side image withih the output micr line (as determined by our recognition engine) appended to the bottom.
- Micr Area Blobs – finally, the micr area blobs that were identified and analyzed within the micr area. These pixel blobs are extracted from the cleaned and smoothed image, and may go through subsequent noise removal steps.

ZIP File Creation

The E13B OCR Results Viewer allows a ZIP file to be created that contains all of the information associated with the current image recognition event. This ZIP file includes the front-side image itself, along with all relevant information that was generated by the recognition process. If desirable and appropriate from your perspective, this ZIP file can be forwarded to X9Ware to allow us to review the results and make potential improvements to our recognition processor.

- The ZIP file contains only information related to a single front-side image that has been inspected and run through recognition. After the information is exported, we ask that you open the ZIP file to review and confirm the enclosed information.
- Be aware that this ZIP file will contain PII (Personally Identifiable Information) that includes the account and routing numbers, along with any information that is pre-printed (like customer name/address) or written onto the front side check image.
- If you decide to forward this ZIP file to X9Ware, we will delete the file as soon as it is reviewed by our technical team. It will not be kept by us any longer than needed to perform this technical review.
- You will need to send this email to X9Ware for our review; it will not be automatically forwarded.
- We will send an email to you when received and then also when deleted.

Summary

The E13B OCR Results Viewer is meant to provide an overview of the steps performed by our recognition engine. It not only provides some useful information, but can also optionally create a ZIP file which details the inputs and outputs from the E13B OCR recognizer.

Tiff Tester

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

Tiff Tester is a convenience tool that allows you to load an external image for inspection and validation. Tiff Tester uses the same internal TIFF image validation rules within X9Assist that are applied to images which are embedded within x9 files. Tiff Tester allows you to quickly and easily perform these powerful validation rules against any TIFF image, when the image is not currently embedded within an existing x9.37 file.

Before launching Tiff Tester from the tool bar, you must first ensure that the desired tiff rules are loaded for your image validation. These rules are loaded whenever an x9.37 file is processed. Tiff Tester will tell you which x9 rules (and hence tiff rules) are currently loaded on the status line in the lower left of the panel. You may need to first validate an x9.37 file (using your desired rules) to ensure that the correct TIFF rules are currently loaded.

Tiff Tag Validation

Tiff Tag validation applies the rules as defined by the x9.100-181 (Tough Tiff) standard to the current image. The Tiff Tester is very straight forward to use:

- Use the LOAD button to load your desired image.
- The image will be displayed in the upper left.
- The tiff tags associated with this image will be displayed in the upper right.
- Any error messages (from the validation) are displayed within the bottom panel.

E13B-OCR Recognition

The Tiff Tester can be used to gain insight into the capabilities of our internally developed E13B-OCR recognition tool. The E13B-OCR Tool Kit is a separate X9Ware product that can be invoked on a stand-alone basis or incorporated into your image applications. Our E13B-OCR Tool Kit is 100% Java and can be run in any environment, with a Java 1.8 minimum. Once an image is loaded, enable the OCR checkbox within the action panel. You will then see a yellow recognition strip added to the bottom of the image, which will contain the E13B recognition results.

The E13B recognizer also allows detailed information to be accumulated and displayed, as insight into the overall recognition process. This is activated using the “?” within the action panel.

These E13B-OCR recognition tools are also built into the X9Assist x9.37 Item Viewer, allowing them to be applied against the images within an existing x9.37 file.

CSV Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The CSV Editor allows the content of a comma separated value (CSV) file to be edited and saved. The editor includes many commonly used tools, which simplifies the task of modifying a CSV file. Most importantly, it allows CSV files that have a variable number of columns to be edited without converting the file to a fixed column format.

Other CSV Editors

Unfortunately, most CSV editors will save a CSV file in a fixed column format, based on the maximum number of rows that are found across the entire file. Although that makes sense for most CSV file usage (especially those that are compliant with RFC4180 which defines a fixed column standard), it does not apply to our CSV usage for X9.37 and ACH files, each of which require a specific number of fields (cells) on each row based on record type. Support for this variable column format is core to our CSV Editor, where trailing non-populated fields on a given row will be gray when they are not present. Editing a variable column CSV file with most other editors may destroy their content, due to the empty fields that might be appended to shorter rows.

Other CSV editors can have various annoying features. For example, MS-Excel often converts numeric values into floating point (“E”) format. MS-Excel also uses unusual and unexpected techniques when adding quote marks around cell values. CSV files saved using MS-Excel are often times unusable. The best CSV Editor we have found is actually LibreOffice, which is quote mark friendly, but still converts files from variable columns to fixed columns.

The bottom line is that our CSV Editor resolves all of these issues by behaving in a manner that is compatible with our CSV file requirements.

Editor Highlights

The CSV editor is a high function tool, since it:

- Supports a variable number of columns on each row.
- Columns are identified using both column numbers and Excel identifiers.
- Implements common row and copy operations.
- Uses extensive highlighting to provide feedback on rows and columns being edited.
- Provides search capabilities to quickly find values within a file.
- Tracks modifications that have been made to the file since it was initially loaded.
- Includes the ability to revert any given row back to it’s initial value.
- Allows files to be saved under their current name or an alternate name.

Trailing-Truncated Fields

Each row within the editor can have a variable number of fields. The number of columns presented by the editor is based on the maximum number of columns found across all input rows.

If a row has trailing fields that are physically omitted from that particular row, then those cells will be grayed-out to indicate that they are not present to indicate they are truncated. These cells are not written when the file is saved.

A trailing field can be marked as truncated (not present) in one of several ways. First by selecting that row using the check box on the left, and then using the truncate command to mark all trailing fields as omitted. This can also be done for a single trailing field by entering a value of '/ into that cell. When '/ is entered, that cell is marked as truncated and will now be grayed out in the editor. The visual use of the grayed-out cells makes it easy to identify these fields.

Trailing fields on a given line can be changed to empty strings using the pad command, which is essentially the opposite of truncate.

A file can be converted from variable columns to fixed columns using the select function (which can be used to select all) and then using the pad command to pad all selected rows. This function should be used carefully, since the concept of variable rows will be lost for this CSV file.

Paste Board

The editor includes a "paste board" which can hold one or more rows or a single column. We refer to this component as the paste board since we do not want it confused with the operating system based clip board; these are entirely different entities.

Any operation that stores data into the paste board will overlay the previous content. The status line (at the bottom of the editor) always informs you as to the current content of the paste board. Highlighters are also included to visually show the data that has been copied into the paste board.

Selecting Rows

Rows are selected as input into our various row oriented command that are available on the status line (at the bottom of the editor). Rows can be selected in one of several methods:

- Column one check boxes (on the left side of the editor) is used to select one or more lines. Selected lines will be highlighted as they are selected.
- A block of rows can be selected by using the check box to select the first and last rows, and then hitting the select button on the status line to select the range. The selected block will then be highlighted.
- The reset button can be used to change all rows/columns to a non-selected status.

Selecting Columns

Columns are selected using the radio buttons that are present at the top of each column. Columns are selected as input to the various column oriented commands (insert, delete, copy, etc) Only one column can be selected at a time.

Editor Commands

The following editor commands exist on the status line, at the bottom editor:

- Insert – is used to insert a new row or new column after the selection.
- Delete – is used to delete the selected rows or column.
- Copy – is used to copy the selected rows or column into the paste board.
- Cut – is used to cut the selected rows or column and store them into the paste board. Those rows will be deleted as part of this operation.
- Paste – is used to paste the content of the rows or column that currently exist in the paste board, after the currently selected row or column. The paste board will be cleared.
- Truncate – is used to mark the trailing fields on the selected rows or column as omitted (not physically present). These cells will become gray and will not be written when saved.
- Pad – is used to pad the selected rows to the maximum number for this CSV file by setting those cells to the empty string.
- Revert – is used to revert the selected rows or column to their original values, at the time that the file was first loaded.
- Select – is used to select multiple rows. A block of rows can be selected by first selecting the first row in the block, then selecting the last row in the block, and finally using the select button on the status line to select the entire block. A column can be selected using the radio button at the top of the column. Highlighting is used to provide visual feedback on the result of the selection and confirmation that it was done as expected.

Saving a File

A modification count is always provided on the status line, in the lower left corner of the editor. This is a count of the number of rows that have been modified.

Save is initiated using the button provided on the status line. A file chooser will be presented allowing the output file to be chosen. Once confirmed, the file is saved and then reloaded into the editor which confirms that the contents are as expected.

Search / Replace

The CSV Editor includes a find function, but does not include a text based search/replace tool. However, you can use typically other more generic text tools (eg, NotePad or NotePad++) when this is needed. These tools have very powerful search/replace capabilities. Because of that, our thought has been that it is not needed in our editor at this time.

Modifying Files Using Import/Export

The CSV Editor can be used in conjunction with our Import/Export tools as a powerful combination which allows files to be modified in a large number of ways. This is accomplished through the following basic steps:

- Export the file (either X9.37 or ACH) to CSV.

- Use the CSV Editor to make changes, which can be such things as remove records, rearrange records using cut/paste, etc.
- Import the modified CSV file.
- Perhaps use Repair based on the final status and the modifications which have been made.

The combination of these tools is especially useful when data is needed for negative testing, where invalid record sequences are needed for specific use cases. This can be fully accomplished using the CSV Editor in a very straight forward manner.

XML Editor

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**YES**	**YES**

The XML Editor allows the content of a properly structured XML file to be edited and saved; access to a defining schema is not required. The editor includes many commonly used tools, which simplifies the task of modifying an XML file. Most importantly, it presents a user interface (UI) that is oriented towards the actual data elements withing the file, and not the Extensible Markup Language (XML) tags that are used to internally define an xml file. This presents numerous advantages over using text editors (eg, NotePad or WordPad) to edit an XML document, since you can concentrate on element content and not the wrapper tags.

Editor Highlights

The XML editor is a high function tool, since it:

- Supports xml documents with a varying number of elements, fields, and attributes.
- Supports xml documents with a varying depth.
- Columns are identified using both column numbers and Excel identifiers.
- Implements many common element level tools including copy and paste.
- Uses extensive highlighting to provide feedback on rows being edited.
- Provides search capabilities to quickly find values within a file.
- Tracks modifications that have been made to the file since it was initially loaded.
- Allows files to be saved under their current name or an alternate name.

Paste Board

The editor includes a “paste board” which can hold the content of one or more rows. Each row in the paste board can be a leaf element (it has no children) or a parent element that owns a sub-tree of elements. We refer to this component as the paste board since we do not want it confused with the operating system based clip board; these are entirely different entities.

The paste board is a great way to copy and paste (or cut and paste) elements from one place within the xml document to another. Just remember that you can copy a sub-tree of elements, which makes this a very powerful function.

Any operation that stores data into the paste board will overlay the previous content. The status line (at the bottom of the editor) always informs you as to the current content of the paste board. Highlighters are also included to visually show the data that has been copied into the paste board.

Selecting Rows

Rows are selected as input into our various row oriented command that are available on the status line (at the bottom of the editor). Rows can be selected in one of several methods:

- A singe row can be selected using the check box for that row.
- Multiple discontiguous rows can be selected by using their individual row check boxes.
- A block of rows can be selected by using the check box to select the first and last rows, and then hitting the select button on the status line to select the range. The entire selected block will then be highlighted.
- The reset button can be used to change all rows/columns to a non-selected status.

Editor Commands

The following editor commands exist on the status line, at the bottom editor:

- Insert – is used to insert a new row after the selection.
- Delete – is used to delete the selected rows or column.
- Copy – is used to copy the selected rows into the paste board.
- Cut – is used to cut the selected rows and store them into the paste board. Those rows will be deleted as part of this operation.
- Paste – is used to paste the content of the rows that currently exist in the paste board, after the currently selected row. The paste board will be cleared.
- Up – is used to move the currently selected row up one slot within the element list.
- Down – is used to move the currently selected row down one slot within the element list.
- Select – is used to select multiple rows. A block of rows can be selected by first selecting the first row in the block, then selecting the last row in the block, and finally using the select button on the status line to select the entire block. Highlighting is used to provide visual feedback on the result of the selection and confirmation that it was done as expected.
- Reset – is used to reset current find status and all row selections.

Saving a File

A modification count is always provided on the status line, in the lower left corner of the editor. This is a count of the number of rows that have been modified.

Save is initiated using the button provided on the status line. A file chooser will be presented allowing the output file to be chosen. Once confirmed, the file is saved and then reloaded into the editor which confirms that the contents are as expected.

Search / Replace

The XML Editor includes a find function, but does not include a text based search/replace tool. However, you can use typically other more generic text tools (eg, NotePad or NotePad++) when this is needed. These tools have very powerful search/replace capabilities. Because of that, our thought has been that it is not needed in our editor at this time.

Use Case Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The Use Case Editor is used to **create** new use case lists. Use case files are stored in folder /x9_assist/lists/useCases. They must be stored in that folder to be available to the Make and Scrub functions.

Use case lists are utilized by several functions within our desktop tools.

- Make can utilize a use case list as the basis to create an input file to Generate. Note that a use case list is the most basic of information that can be processed by Make, since Make can take many more columns (data elements) than are available in a use case list.
- Scrub can utilize a use case list as part of sanitizing the data in check detail and return detail records.

Use case lists consist of the following four data elements:

- ABA
- Account number
- Process control field (consumer check serial number and possible transaction code)
- Auxiliary OnUs for commercial (long) checks

Use Case Creation Alternatives

The Use Case Creator is one of several tools that you can use to create a new use case file.

You need to consider your alternatives as to how you want to create and maintain your use case files. Here are several alternatives for the creation of these lists:

- Use the supplied Use Case Editor tool
- Use our Export tool to build a list from your production x9 files
- Use MS_Excel (or a similar tool) to create and maintain a list in XLS or CSV format (Excel XLS files must be saved as an Excel 97-2003 workbook)
- Use other tools (like PERL, GREP, etc) to create use case files from your test beds
- Develop your own applications (C, Java, etc) to extract data from your test systems and put the test information into our required use case format

Take special care when you use this function that you do not mistakenly overwrite an existing file that you do not want replaced.

Options on the input panel allow you to:

- Either assign the same ABA to all entries or select the ABA randomly from a list. You can utilize an ABA list provided with X9Assist or you can build and maintain your own ABA lists for your unique purposes.
- Select where you want the check serial number placed. The check serial number can be up to ten digits long. Typically, any check serial number longer than four digits must be placed in the Auxiliary OnUs field. You can indicate that the check serial number should be placed in the Process Control field, then you should specify parameters such that the serial number does not exceed four digits. You can alternatively indicate that the check serial number be placed in the Auxiliary OnUs field, or should be placed in the appropriate field based on the actual length of the check serial number.
- Select the ABA list to be used if you are randomly selecting ABA numbers from a list.
- Select an account number ModCheck routine that you want to use to direct the generation of account numbers. You can use routine definitions that are installed with X9Assist or you can use the ModCheck Editor to define your own routines.
- Indicate the number of account numbers to be generated.
- Indicate the range (low to high) from which account numbers will be issued.
- Indicate the range (low to high) for check serial numbers to be randomly issued.
- Optionally indicate a prefix to be appended to the Process Control field.
- Optionally indicate a suffix to be appended to the Process Control field.

Use Case List Formats

Use case lists must be stored in either Excel or CSV format. You need to consider which of these formats will be easier to work with based on your specific requirements.

If you are creating the use case file automatically using your own tools, then using the CSV format is probably your best option.

If you are editing the use case file using tools such as Excel, then for ease of use you should maintain the use case file in XLS format. These native Excel formats can be read directly by any function within X9Assist that would otherwise process a CSV.

When building and maintaining a use case list in CSV format, you will need to be aware of and follow some general formatting requirements.

Per CSV file standards, numeric fields do not need to be enclosed in quotes. Since these are all numeric fields, there is no absolute requirement to enclose the field values in quote marks (they are optional). However, missing fields (those that are not present) are indicated through the use of two consecutive quote marks.

Your quote mark usage can be single quote (‘) or double quote (“) but must be used consistently throughout your entire file. Field separators can be either commas or tab characters.

Configuration Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The Configuration Editor is used to define the various x9 configurations that can be used for x9 file validations. Each configuration is a logical group of rule files that, when collectively taken as a group, define the validations that will be applied against an x9 file.

X9Assist is installed with a predefined list of configurations. List standard list can then be changed using the Configuration Editor based on your specific requirements. This is especially important when you are adding your own x9 rule definitions and hence must construct a configuration to utilize your custom x9 rules. Every configuration consists of the following attributes:

Attributes	Attribute Content and Purpose
X9 rules file	Contains a list of the rules that are applied to the x9 records and fields within the x9 file. These rules are used to generate x9 related errors.
Tiff rules file	Contains a list of the rules that are applied to the tiff images within the x9 file. These rules are used to generate image related errors.
Messages file	Contains a list of the error messages and their associated severity levels. These rules are used to assign the severity level of the errors that are generated during the validation process. Each error has an assigned severity level of Error, Warn, Info, or OK.
Entry Type	Defines the configuration as either “System” or “User”. System entries are automatically defined within the X9Assist environment and cannot be modified by the user. User entries can be created using the Configuration Editor and will be automatically retained across user sessions and X9Assist release installations.

Each configuration is displayed with a “status” column, which provides the current validation status of the components with that configuration. If the configuration is displayed with a check mark, then all of the components within that configuration are present and validated.

Entries can be deleted by selecting them (via their check box) and then pressing the delete button. System entries cannot be deleted.

This table must be saved for your changes to be effective. Results will be saved to xml/config.xml.

System Entries versus User Entries

A system entry is one which is created internally and added to this table. The content of system entries may change from release to release. The editor will allow user entries to be modified or deleted, while system entries are not allowed to be changed.

Load Identifiers

The Configuration Editor load identifier column provides information as to the location and time stamp for each physical component. The location can be in one of three places:

- The launch folder (where X9Assist was loaded from) is identified as “L”.
- The home folder (within user documents) is identified as “H”.
- Otherwise, the load identifier can be blank, which indicates that the component was not found in either of these two locations but instead defaulted to the internal resources definition that is defined within the SDK itself. This is not a typical configuration for X9Assist, since our rules are distributed with that product and are installed in the launch folder.

By convention, all rules should be stored in the launch folder (they should not be placed in the home folder). This is because rule parsing is dependent on the associated code that is implemented within the current release. Locating rules in the home folder would potentially make them available to more than one release of X9Assist, which would result in unpredictable results and most probably program aborts.

Construction of the Configuration List

X9Assist loads the internal configuration list during session initialization as follows:

- All system entries are internally populated to the configuration list based on the current release definitions. When you install a new X9Assist release, this may result in a new configuration list entry being automatically added, or it may alternatively result in an obsolete entry being removed.
- All user entries are then automatically merged from the config.xml definition. Only user entries are added during this merge process. System entries are not needed since they are already been populated. If you need to modify a system entry, then you should create a user entry with the same name, which will then overlay the system entry with that same name.

Default Session Configuration

The configuration editor allows assignment of the default session configuration. When this configuration has been set, it will override the standard use of our auto-detection process and instead will be used for validation whenever a new file is opened. Other configurations can still be made available for use, but they must be invoked manually from the tool bar.

Standard Configurations

X9Assist and the X9Ware SDK support the following x9.37 industry specifications:

Standard	Description
DSTU	Introduced in 2003 as part of the Check Clearing for the 21st Century Act, the DSTU is the original and most commonly used image exchange standard which establishes the basis for U.S. check image exchange between two financial institutions. The DSTU continues to be widely used since it is the most basic of standards and it generally accepted throughout the industry. X9Ware fully supports the X9.37 DSTU standard.
X9.100-187	X9.100-187 was introduced in 2008 as an updated replacement for the DSTU. It is used as the basis of the specifications that have been adopted by many financial institutions as a DSTU replacement, and clarifies many of the gaps that existed within the original standard. X9.100-187 standard was updated in 2013 and 2016. Each of the X9.100-187 standards has an associated Universal Companion Document (UCD) which is an extension that contains commonly accepted by financial institutions and third party processors. X9Ware provides support for X9.100-187-2008, X9.100-187-2013, X9.100-187-2016, X9.100-187-2008-UCD, X9.100-187-2013-UCD, and X9.100-187-2018-UCD.
X9.100-180	X9.100-180 was published in 2006 with the intention to address many of the business requirements that were identified with the DSTU. The biggest desirable change was the ability to support credits, along with separate debit and credit totals in trailer records. Unfortunately, the various record types and fields within X9.100-180 became so different from the DSTU that adoption of this standard became extremely difficult. The X9.100-187 UCD was the industry response to X9.100-180, allowing specific needs to be addressed in more straight forward ways. As a result, X9.100-180 is largely unused by most financial institutions. X9Ware provides support for X9.100-180-2006 and X9.100-180-2016.
CPA 015	Canadian Payments Association Standard 015 (CPA 015) is an extension to X9.100-187 and defines how image exchange has been implemented within Canada. This standard addresses currency requirements in support of CAD and USD and supports cross border exchange between banks. X9Ware provides full support for the CPA-015 standard.
Cross Border	X9Ware has implemented various extensions to the X9.100-187 standards in support of cross border exchange. These extensions are limited to accepting routing numbers in a variety of formats including 4-4 and 5-3 as required by Canadian financial institutions.
FRB	The FRB provides electronic check image exchange services, based on the DSTU which is fully supported by X9Ware.
EndPoint Exchange	Endpoint Exchange provides electronic check image exchange services based on the DSTU which is fully supported by X9Ware.

Standard	Description
ViewPointe	ViewPointe provides electronic check image exchange services based on the DSTU which is fully supported by X9Ware.
SVPCo	SVPCo provides electronic check image exchange services based on the DSTU which is fully supported by X9Ware.

Configuration Names

Configuration names are used internally to identify the standard level. This is a basic requirement must be taken into consideration as you assign names to user defined configuration. This is accomplished through identification strings that must be embedded within the configuration name, as identified by this table:

X9 Specification	Must Contain the Following Identification String within the Assigned Name
DSTU	2003
X9.100-187	100-187
X9.100-187-2008	100-187-2008
X9.100-187-2013	100-187-2013
X9.100-187-2016	100-187-2016
UCD	UCD
X9.100-180	100-180
X9.100-180-2006	100-180-2006
X9.100-180-2013	100-180-2013

Rules Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

Rules Editor is an advanced topic that allows rules files to be viewed, modified, and saved. The resulting xml file can be utilized by the Configuration Editor to build custom validation rules.

Rules are used to define the records and fields for a given specification. These rules define the record types that are accepted and the list of fields that constitute that specification.

X9Validator and X9Assist have some of the most advanced and flexible validation capabilities that you will find within the industry today. The X9Rules Editor allows configurations to be customized based on specific requirements for records, fields, and their associated validation rules.

The rules panel has two tabs which are used within the editor:

- Controls – used to provide the values for the control values that are to be assigned to this set of validation rules. The editor allows you to indicate if a given control is to have a value assigned (included in the output xml) or allowed to default (excluded from the output xml). This is especially important for extension documents, where a given control value should be inherited from the higher level document. You only need to attach a specific control value to the current x9rules definition when you need to explicitly override the assignment default, which may come from the higher level rules document.
- Records – used to specify high level controls along with the list of records that are defined within this definition. The controls are essentially attributes that are utilized by the validation engine. The attributes themselves are largely self explanatory. Similarly, the list of records defines all of the record types that will be accepted by this definition.
- Fields– used to enter a detailed table of fields that are associated with this rules definition. These entries will vary based on the specification that is being implemented, per those requirements.

The Rules Editor can be used to modify an existing set of validation rules, and then either update those rules or save them to an alternative xml file name.

Please refer to the X9 Rules topic within the SDK User Guide for more detailed information about this xml file and content.

Tiff Rules Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

Tiff Rules Editor is an advanced topic that allows tiff rules files to be viewed, modified, and saved. The resulting xml file can be utilized by the Configuration Editor to build custom validation rules.

X9Validator and X9Assist have some of the most advanced and flexible tiff image validation capabilities that you will find within the industry today. The Tiff Rules Editor allows configurations to be customized based on specific requirements for tiff tag validation, EOFB validation, and pixel density associated with IQA (image quality) testing.

The tiff rules panel is divided into two sections:

- Controls – which is used to specify high level controls. These controls are essentially attributes that are directly associated with tiff image validation. The attributes themselves are largely self explanatory.
- Tiff Tags – which is used to enter a detailed table of tiff tags as used by the tiff image validation engine. These entries identify detailed data about individual tiff tags, as per the requirements defined within the x9.100-181 image exchange standard.

The Tiff Rules Editor can be used to modify an existing set of tiff rules, and then either update those rules or save them to an alternative xml file name.

Please refer to the TIFF Rules topic within the SDK User Guide for more detailed information about this xml file and content.

Mapping Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The binding process is designed to allow incoming files to be validated per specific user requirements, where rules are appropriately applied based on the type of file that has been received. The design goal is to attempt to do all of this on an automated basis based on header record inspection. X9Assist refers to this processing as ***binding***, with the attempt to automatically the correct rules to specific file content.

The x9 standard level within the file control header identifies the x9 specification that was used to encode that specific x9 document. Unfortunately, due to a goal to maximize file compatibility across the various x9 specifications, the x9 standard level is field can be insufficient in many situations to fully and accurately identify the x9 standard to be applied to a given x9 file. Binding and mapping provides automation to supplement this decision process.

Across many financial institutions, the combination of Origination and Destination Routing can be used to fully define the validation rules that are to be applied to the file. X9Assist allows this definition to be by Routing, but also allows the definition to be done on a generic basis for Origination Routing, Destination Routing, or both.

For example, you can define the configuration to be applied to a specific Destination Routing regardless of origination. Or you can define the configuration to be applied to a specific Origination Routing regardless of destination. In many environments, the configuration can be assigned from the file header Standard Level and UCD Indicator, with no need to drill down to the Origination and Destination Routing levels.

Through use of the Mapping Editor, you can provide automated instructions to the binding process which allow you to define overrides (exceptions) that are based upon the Origination and Destination Routing from the x9 file header. This allows you to ensure that X9Assist will automatically use the correct configuration file based on your specific requirements.

System Entries versus User Entries

A system entry is one which is created internally and added to this table. The content of system entries may change from release to release. The editor will allow user entries to be modified or deleted, while system entries are not allowed to be changed.

Use of the Mapping Editor

The Mapping Editor defines the rules which are used by the Binder to automatically assign incoming x9 files to system defined configurations. This automated bind process is largely accomplished based on the contents of the type 01 file control header.

The following file control header fields are used to drive the automated bind process and can be specified using the Mapping Editor:

- Origination Routing
- Destination Routing
- Standard Level
- UCD Indicator

Each entry within the mapping table is defined as either a system entry or a user entry:

- System entries are automatically added (internally) by X9Assist.
- User entries can be added and maintained by the user. These definitions can either supplement or overrides to the system entries that are internally created.

By first establishing configurations and then defining your mapping to those configurations, you can provide information to X9Assist to automatically use the appropriate x9 and tiff rules when applying validations to an x9 file. For example, if you have a specific and consistent destination RT that is used for image exchange files, you can easily direct X9Assist to automatically apply the correct edits to those files.

These fields are concatenated and used to construct a logical key which must be unique for each entry added to the Mapping Editor.

The status column provides an indication if the defined configuration has been located and that all rules associated with the configuration exist.

Entries can be deleted by selecting them (via their check box) and then selecting delete.

This table must be saved for your changes to be effective. The reset button can be used to restore to your last save point. Results will be saved to xml/map.xml.

Mapping Entry Load Sequence

The mapping file is loaded as follows:

- The process begins with an empty table.
- User entries are first loaded to the table from the “map.xml” file. All user entries from this file are sequentially loaded to the table (note that system entries on “map.xml” are ignored since they will instead be reconstructed and added at load time. The key for each user entry (origination routing, destination routing, standard level, and UCD indicator) must not exist within an existing table entry for the new entry to be loaded.
- Standard system entries are then dynamically created and internally added to the mapping table. Each system entry is loaded only when it's mapping key (origination routing, destination routing, standard level, and UCD indicator) does not already exist within the

table. This means that a system entry will not overlay and will thus never replace a user defined mapping entry.

Hierarchical Search

The search process against the mapping file is one using specific values followed by a generic value of “all”. For example, the search is done using the origination routing, and if an entry is not found using that specific value, then it is substituted with a value “all” and the search is repeated. This hierarchical search is performed until a matching mapping key is identified. In a worse case scenario, a key of “all/all/all/all” will used until on a default basis to identify the configuration to be used for x9 bind. This means that an “all/all/all/all” entry must always be present.

X9 System Configuration as defined in Program Options

You can define the configuration that will ***always*** be used in program options. This is done from the “X9” tab using the “X9 System Configuration” setting.

The default is for this to be ***system assigned***, which means that the configuration will be determined from the contents of the x9 file header. X9Assist will then use either X9.37 or X9.100-187 + UCD rules to apply validations based on the contents of the file header.

Through use of program options, you can override this process to always use a specific configuration file. For example, if your standard is to always use the newer X9.100-187 + UCD validation rules, then you can define this in your program options. In this way, binding is no longer dynamic based on the contents of the file header, but instead will always be done using your specified configuration.

Endpoint Validation

X9Assist has the ability to validate the Origination and Destination routing numbers that are present in the file header. Use of this feature is optional, which means that you can allow all such routing numbers to be received and accepted. You can also decide to validate the Origination Routing, the Destination Routing, or both of these fields.

This validation includes the ability to define an endpoint with start and end dates which define when x9 files will be accepted. Depending on your application environment, it may be possible to extract and even synchronize the endpoint table from your electronic cash letter system and make that available to X9Assist in support of this validation process.

Endpoint Editor

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**YES**	**YES**

The Endpoint Editor is an optional facility that allows origination and destination routing numbers to be defined within X9Assist. You then define the following attributes for each routing number that appears within the endpoint table:

- Description (bank name, etc)
- Client type (you must choose from the drop down list which is provided)
- Start date (this is optional and is the beginning processing date for this routing)
- End date (this is optional and is the ending processing date for this routing)
- Origination (indicates if this routing number can be an origination endpoint)
- Destination (indicates if this routing number can be a destination endpoint)
- Active (indicates if this routing number entry is currently active)

System Entries versus User Entries

A system entry is one which is created internally and added to this table. The content of system entries may change from release to release. The editor will allow user entries to be modified or deleted, while system entries are not allowed to be changed.

X9 File Validation Using the Endpoint Table

During file validation, the origination and destination routing numbers from the type 01 file header are compared against the system defined endpoint table. If the incoming routing (origination or destination) has been predefined in this table, X9Assist will then ensure that the context is allowed and that the endpoint is active based on possible start and end dates (when assigned). This table can be used to ensure that files are only received from predefined originators and are only created to be sent to predefined destinations.

An entry of ten (10) nines is a generic entry that represents an endpoint for “All other banks”. This entry is used to indicate that files of undefined origin and/or destination are to be accepted for x9 file origination, destination, or both.

- With the generic entry in place, X9Assist will create informational messages when the origination and/or destination endpoint has issues (for example, is not currently active based on dates, is not marked as as a file origination endpoint, or is not marked as a file destination endpoint). If the generic entry is marked to accept for both origination and destination, X9Assist then accepts files both from and to any endpoint that is not defined within the endpoint table. You may also, for example, mark the generic entry as applicable to destination endpoints only. In that case, files will be accepted for undefined destination

endpoints (those not predefined in the endpoint table). However, X9Assist will not automatically accept incoming x9 files with undefined origination routing numbers.

- Without the generic entry in place, X9Assist requires that all incoming origination and destination endpoints be predefined and will generate messages when file header routing numbers are either not defined, not appropriate marked for their associated usage context, or are not currently active.

This table must be saved for your changes to be effective. The reset button can be used to restore to your last save point. Results will be saved to xml/routing.xml.

Message Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The Message Editor is facility can be used to edit and maintain error message file(s) on an interactive basis. The editor is used when there is a need to modify the standard error messages that are issued during file validations. Messages are assigned an internal identifier that is used to as a lookup mechanism to associate a specific error scenario with alternate message text.

We have given considerable consideration as to how how error messages have been structured along with the text that has been assigned to each individual message. Although this facility has been created to allow messages to be customized, we would rather ask that you engage us when you have issues with any specific error message condition. We would rather make improvements to the overall environment whenever that possible.

Within the overall environment, there are two types of message files:

- System – which is an internally defined set of messages that are used on a default basis. Since this default definition is defined internally, this complete message set does not need to be defined in the message table.
- User – which is an externally defined set of messages that override the system message file. Since these are overrides, you only need to define those messages where you want the assigned attributes to differ from the standard messages. When you save a user message file, the editor will automatically remove all messages where the attributes (severity and description) match the standard message. In this way, user message files will only contain override messages.

For example, if you want to change the severity or text that is associated with two system messages, then you can create a user defined messages file that contains just those two messages that are to be modified. The record viewer (which is shown in the lower left corner of the dashboard) can be used to identify the message identifier that is associated with the error messages that are being thrown against any particular error. This panel includes a slider, which can be used to shift to the rightmost columns, where the error message name will be shown. This is the message identifier which can be used for message overrides.

This message file will contain only the two override messages. These overrides will be applied as overrides to the standard set of messages. This design allows your user message set to be focused only on your modifications, and also makes it fairly independent of new system releases, since it only contains modifications, and does not need to contain the entire message set.

Message File Locations

As noted, all standard messages are defined internally and are not needed in an external message xml file. Because of that, the standard messages (with unchanged severity and description) will be

automatically removed by the editor when a user message file is saved.

In support of both documentation and transparency, the standard message file (which is named systemMessage.xml) is provided in the distribution and is included in the program launch folder, within folder / rules / messages /.

User message files can be stored in this same folder, or can alternatively be stored in the user's home folder, which would typically be / UserId / documents / x9_assist / rules / messages /.

User Message Files

The Message Editor can be used to edit the system messages file, apply modifications, and then save that file to / documents / x9_assist / rules / messages / using a name of your choice. When you save a customized user message file, the editor will remove all standard messages (those where severity and message description have not been changed). Those messages are not required since they will automatically default to the standard setting. This xml file can then be incorporated into a custom configuration using the Configuration Editor.

Configurations

User message files can be created and then used by the Configuration Editor to create customized configurations for your organization. Configurations are identified as either system or user, where the user configurations typically represent variants to the standard specifications. A user configuration can be created using a customized message set and then distributed within your organization for generate usage.

Message Attributes

The following message attributes are required:

- Identifier – is a drop down box where the selections are internally defined. This field is required to identify the specific message which is being referenced.
- Field – is a drop down box which allows the message override to be tagged to either all fields or to a specific field.
- Description – is a text box where the associated error message text must be entered. This text will be included when the message is thrown during the file validation process.
- Severity – is a drop down box where the selections are internally defined. This value identifies the severity level to be assigned to the message.
- Format – is a drop down box where the selections are full, sparse, and plain. This value identifies the format level assigned to the message. Default is full which indicates that all message fields will be populated in the resulting message. A value of sparse assigns a small set of critical fields, while plain populates the message text only.

Basis Documents

User message files are by their very definition an extension of the system messages xml file. This means that they are essentially overrides to that standard definition. In this same manner, it is also possible to have multiple levels of user message files, where one message file is an extension of another. For example, messageFile2 is an extension of messageFile1, and then logically messageFile1 is an extension of the system messages file. In this usage pattern, messageFile2 has a basis of messageFile1 and overrides (and extends) the messages as defined within that basis document.

Sample Message File

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<errorMessages>
   <copyright>X9Ware LLC 2012-2018</copyright>
   <company>X9Ware LLC</company>
   <release>R4.01</release>
   <buildDate>2019.05.21</buildDate>
   <timestamp>20190522_110531_079</timestamp>
   <basis></basis>
   <messages>
      <error>
         <id>missingFileTrailer</id>
         <sev>Ignore</sev>
         <desc>Missing file trailer record</desc>
         <recordType>0</recordType>
         <fieldName></fieldName>
         <pattern></pattern>
         <format>Plain</format>
         <type>User</type>
      </error>
   </messages>
</errorMessages>
```

Message Editor Functions

The following functions are provided by the message editor:

- Insert – inserts a row into the table (either after a selected row, or at the beginning).
- Delete – delete all selected rows.
- Find – find a user entered string.
- Save – save the table.

This table must be saved for your changes to be effective. The reset button can be used to restore to your last save point. Results will be saved to the message file of your choice.

ModCheck Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The ModCheck Editor is an optional facility that can be used to define account number modulus routines, which are typically referred to as self check routines. The ModCheck Editor is a generic facility that is capable of generating the ModCheck values utilized by most financial institutions.

ModCheck supports the two most common algorithms, with varying account number lengths, any divisor (although they are typically 10 or 11), any weights, and any required remainder.

ModCheck Routines

For each self check routine, you must specify the following information:

- Routine name.
- Algorithm type:
- Sum Of Products indicates that each digit is multiplied by its corresponding weight, with the result total then added to the running accumulation. For example, if the account number digit is 8 and the associated weight is 7, then the product of 56 will be added into the running accumulation.
- Sum Of Products Digits indicates that each digit is multiplied by its corresponding weight, with the sum of the multiplication digits then added to the running accumulation. For example, if the account number digit is 8 and the associated weight is 7, then the product is 56, and the sum of 5+6 = 11 will be added into the running accumulation.
- Length, which is the number of account number digits being checked.
- Divisor, which is used for the division to get to the remainder. The divisor is typically 10 or 11 but can vary depending on the algorithm.
- Expected remainder (typically zero, but which can vary depending on the algorithm).
- Weights, where each weight is associated with a corresponding column (digit) in the generated account numbers. The number of weights must equal the account length field.
- Description essentially your comments to better describe the purpose of this routine.

System Entries versus User Entries

A system entry is one which is created internally and added to this table. The content of system entries may change from release to release. The editor will allow user entries to be modified or deleted, while system entries are not allowed to be changed.

Routing List Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The Routing List Editor allows you to create lists of sequentially assigned routing numbers for one or more US Federal Reserve districts. These lists will be created with the appropriate MOD10 check digit. These ABA lists can be used by various functions such as the Use Case Editor (for the creation of random use cases) or Generate (for the creation of random endorsements).

The ABA List Editor allows you to select one of four creation options:

- Create for a single FED district that you can specify
- Create for all FED bank districts
- Create for all FED thrift districts
- Create for all FED electronic districts

You can specify the number of entries to be created for each district. For example, if you specify that 100 entries be created for each district and then you select all bank districts, then a total of 1,200 entries will be created.

You can also specify the beginning range within each district to be created, with the default being 777000. The Routing List Editor will create each number as the district, the next sequential number within the range, and the appropriate MOD10 check digit.

When you select Create, you will be prompted for the output CSV file to be created. Routing number lists should be stored in the / x9_assist/ lists/ routing / folder to be used by other functions within X9Assist.

Credit Table Editor

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**NO**	**NO**

The Credit Table Editor is an optional facility that is used to edit POD credit tables on an interactive basis. Credit tables are xml files which identify type 25 items that are credits (not debits) based on current capture rules. Each set of x9rules can be associated with a credit table via the x9rules definition. Refer to the SDK User Guide for more information on this topic.

Credit Table Editor Overview

Credit tables are an advanced feature that can be used to simulate a POD capture environment. In this scenario, an Image Cash Letter (ICL) contains logical transactions which consist of credits offset by debits, where all items are identified using type 25 item detail records. Each transaction can consist of one or more credits offset by one or more debits. ICLs are often in this format, where the electronic deposit contains a deposit ticket offset by items. In fact, depending on the environment, ICLs may actually contain multiple deposits, either to the same or different accounts at the depositor's financial institution. Using this X9Assist feature adds value in several ways:

- Once a type 25 record is identified as a credit, it allows X9Assist to balance the individual transaction. A validation error will be thrown when a deposit is out of balance (the credit total amount does not equal the debit amount).
- In addition, another validation error will be thrown when the overall file is out of balance (the file credit total amount does not equal the file debit total amount).

The credit table is defined separately from x9rules. The x9controls section uses parameter "t25ClientCreditTable" to define the credit table name. This name can be provided in either an absolute or relative basis. When relative, the file must be defined within either the launch or home folder, within / rules / tables / . When absolute, the credit table can be stored within a folder that is convenient for updating. The use of an absolute location allows the definition within x9rules to be provided just once (as a pointer) with the credit table then updated externally as needed.

Credit tables can be defined using a client identifier, which is constructed from the ECE institution and destination routings from the cash letter header. This approach allows a single credit table to identify credit information for multiple clients within your capture environment.

The client table (located within the credit table) then defines one or more credits using the payor routing and transaction code from the type 25 record. Each credit configuration identifies routing(s) and transaction code(s). These are specified on a WYSIWYG basis. If both nine digit and 4-4 routings are to be accepted, then multiple routings must be provided.

For a given routing, the transaction code is optional. When the transaction code is omitted, then the routing is dedicated to credits, which means that no transaction code is required. Alternatively, the routing can be shared between debits and credits. In that situation, a credit is identified when the transaction code (from the MICR OnUs value) matches a provided value.

The client table may become a large document. X9Assist includes the Credit Table Editor which

allows the credit table to be updated on an interactive basis. The credit table can also be constructed on an automated (programmatic) basis from POD (capture system) tables. Given the potential size and complexity of the credit tables, it is not recommended that they are edited using XML text editors.

Credit Table Editor Functions

The following functions are provided by the credit table editor:

- Insert – inserts a row into the table (either after a selected row, or at the beginning).
- Delete – delete all selected rows.
- Duplicate – duplicate the selected row.
- Find – find a user entered string.
- Save – save the table.

This table must be saved for your changes to be effective. The reset button can be used to restore to your last save point. Results will be saved to the credit table name of your choice.

HeaderXml937 Editor

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**YES**	**YES**		**YES**	**NO**	**NO**

HeaderXml937 Editor is an editor that allows the HeaderXml parameter files used by the X9Utilities "-write" command to be viewed, modified, and saved. These xml parameter files define the x9.37 attributes that are used by "-write" to generate output files.

HeaderXml937 Editor is available as part of our X9Validator and X9Assist desktop tools. These xml parameter files define the x9.37 attributes that are used by the X9Utilities "-write" function to generate output files.

This editor makes the process of creating and updating those parameters much easier than the alternative of using a simple text editor (eg, NotePad or NotePad++). Using our HeaderXml937 Editor eliminates the xml document errors that can often result from that alternative process. This editor allows you to concentrate on content and not all of the technical intricacies that are associated with an XML file.

There are a substantial number of fields defined within the HeaderXml definition, which can add to the complexity of both entering and updating these fields. To make things a bit easier, all fields have been grouped based on function, using tabs which are located in the right-most column of the editor. These groups bring related fields together onto a single panel. This makes it much easier to find any given parameter and allows you to easily view all of the parameters for that same topic.

Each of the field panels have several right-side columns that provide additional insight into the fields associated fields.

- The far-right column is formatted with [nn] where "nn" is the maximum length of this data field. The editor will begin to beep (as an error indicator) if you attempt to overfill the capability of any given field
- The second column from the right provided information as to how a given field is used, or where the entered data will be populated, which can be very useful information when creating and modifying these parameter files.

The following tabs are available within the editor:

- Structure – defines high level fields that are associated with the overall file.
- File Headers – defines fields that are associated with the type 01 record.
- Cash Letter Headers – defines fields that are associated with the type 10 record.
- Items – defines fields that are associated with the type 25 record.
- Credits – defines fields that are associated with the type 61/62/25 credit records.

- BOFD Addenda – defines fields that are associated with the type 26 (forward presentment) or type 32 (return) records that are attached to all items.
- Secondary Addenda – defines fields that are associated with the type 28 (forward presentment) or type 35 (return) records that are attached to all items.
- Image View Detail – defines fields that are associated with the type 50 record.
- Image View Data – defines fields that are associated with the type 52 record.
- MICR Symbols – defines the control characters that are to be used when building MICR line data from the underlying component fields.

The following functions are provided on the action line at the bottom of the editor panel:

- Cancel – exits the editor; anything that has been saved will remain in that state and will not be undone.
- Load – allows a new file to be loaded. The current content within the editor will be replaced with the chosen file; the current content will be lost.
- Save – allows the current editor contents to be saved to an output XML file.
- Reset – resets the editor to a default status.

Template Editor

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
NO	**NO**	**YES**		**YES**	**NO**	**NO**

Check image creation within X9Assist, as utilized by Make/Generate, can be extended with your own custom artwork. This includes modifications to our predefined templates (for example, changing a font size or definition) or adding your own templates. Image formats are defined within predefined categories, which are retail (for wallet side checks), business (for commercial size checks that include the auxiliary OnUs field), and credits. These category definitions may be extended in the future based on user requirements.

The Template Editor allows the image templates used by Make/Generate to be viewed, modified, created, and saved. Template definitions are xml files stored in the / documents / x9_assist / xml / templates / folder. Overall, this process can become complex, and use of this tool should be considered as an advanced topic. We recommend that you frequently backup the xml template definitions when the Template Editor is being used.

The editor provides functionality in two areas. First is that it allows the x/y positions of template fields to be updated based on their visual placement within the template. Without this, the process of adjusting positions is very complex, since you cannot easily determine the overlay positions of the various template fields. Second is that the editor eliminates the need to manually update the associated xml files needed to support templates. Without the editor, these xml files must be updated on a manual basis using tools such as NotePad, NotePad++, or WordPad.

This editor makes the process of creating and updating those parameters much easier than the alternative of using a simple text editor (eg, NotePad or NotePad++). Using our Template Editor eliminates the xml document errors that can often result from that alternative process. This editor allows you to concentrate on content and not all of the technical intricacies that are associated with an XML file.

Functionality

The editor allows existing templates to be viewed and modified. It also allows new templates to be added or deleted. All needed xml files (including templates.xml) are updated.

The editor also includes a rudimentary paint tool. We realize that the paint tools are very basic, but there are enough tools to allow basic templates to be drawn. The editor allows an existing template to be saved with an alternate name, allowing it to be the basis for new work. This is the only available method for creating a new template.

The paint tools include basic shapes including line, rectangle, and text, along with undo and redo. These shapes can be drawn onto the underlying image, and will be burned into the template image when saved. The paint tools are an attempt to simplify your template development work. They are

not envisioned as the end-all that would completely eliminate the need to use other tools such as MS-Paint or Gimp.

The editor displays the currently loaded template image within a drawing canvas in the upper left panel. Mouse coordinates are displayed on the toolbar as the move is moved within the panel. Our drawing actions are located on the toolbar, and you should read through the popup help that is associated with each functional icon.

The editor also displays the fields that have x-y coordinates for this template. These fields are shown within the lower panel, where the data can be modified. For example, to move a template field to a new position on the drawing canvas, move the mouse to where you want the field to be positioned. Make note of the mouse x-y coordinates and then enter those positions within the field table. You will see that the field is then moved accordingly within the canvas.

Fields can be displayed using either an X9Assist font or a System font. The X9Assist fonts are internally defined and always available. The System fonts are dependent on the fonts that are available within the system font directory for the current workstation. You need to be careful when using System fonts, since these fonts are not permanent; they may come and go.

As part of saving a template, the image will always be written in black-white TIFF, regardless of the original format. This is an image exchange (x9.100-181) standard.

Delete formats WITH CARE. You should never delete a system supplied format, since X9Assist is dependent on the existence of these templates.

Tools

The editor supports the following tools:

1. Text – adds specific text where the font and size can be selected.
2. Line – draws a straight line, which can be dragged to size.
3. Rectangle – draws a rectangle, which can be dragged to size.
4. Draw – allows free form text to be drawn, using the current brush size.
5. Erase – allows image pixels to be cleared, using the current brush size.
6. Copy – copies the shape that is current positioned under the mouse.
7. Remove – removes the shape that is current positioned under the mouse.
8. Move – moves the shape that is current positioned under the mouse.
9. Resize – resizes the shape that is current positioned under the mouse.
10. Undo – repetitively undo the last action.
11. Redo – repetitively redo the last action.
12. Save – saves the current template to the name and category identified on the right-side panel. The current template can be saved as an alternate name, which can then be edited.
13. Delete – deletes the current template.
14. Reset – resets the editor.
15. Cancel – exits the editor.

Manual Updates (not using the Editor)

If you decide to update templates on a manual basis (not using the editor), then the following steps must be followed when adding check artwork:

- Use a third party paint tool to create your check artwork. You can look at our existing formats in the x9assist/images folder to get a good idea of how the current check image formats have been designed.
- A key part of your design is to ensure that MICR band (the bottom 9/16 of an inch) remains blank at the bottom of the image.
- When creating your check formats, you must consider the size (especially the length) based on your desire to support the Aux OnUs field which is required on business checks. Your check formats must also be defined within the minimum and maximums per industry standards.
- Our check formats were designed and drawn using Photoshop which allows a very clear and sharp image to be created. You could do the same using other tools such as GIMP or MS-Paint. Depending on your requirements, this can become a somewhat complex task and does require a good working knowledge of the image design tools that you select. MS-Paint is especially good for adding overly text to an existing image.
- The images should be saved in black-white format, either as TIFF or PNG. We have found this a bit complex using some of the above mentioned image tools, but we have found this conversion much easier using IrfanView64. Obviously, your choice of tools is up to you.
- The template images must be saved at 240 DPI.
- Each format has a front image and an optional back image.
- Once the images are created and in black-white format, they should be store your created image format(s) in folder: / Program Launch / X9Ware LLC / X9Assist Rx.xx /images/. You can take look at our images to get a feeling for how they were saved.
- All check formats are defined in multiple places, as follows:
    - Each format is defined in “templates.xml” within folder / xml / templates /. This is an index of all available templates and defines their associated category. When you create a new entry within the templates xml file, make sure it is marked as “User” and not “System”. This is because the system entries are internally defined, which means that it is important that your additions be marked as user entries.
    - Each format is then fully defined within a check format xml file that must be stored within the appropriate category folder.
    - Finally, the associated images must be stored in folder: / images /.
- You will need review these existing xml files and get a good understanding of how check formats and their associated fields are specified. You will need to use an editor and add your new formats and fields to this xml definition. For each logical element within your check image format, you will provide the font, x/y coordinates, and length. Make sure you store this updated file in a safe location (where it is backed up) since it may be updated when you install the next X9Assist release.
- Determine what fonts you want to use for the various fields that will be drawn on top of your image format. You must also provide the initial font size to be used for the drawing process. If the drawn text will not fit within the available space, X9Assist will

progressively reduce the font size until the text will fit. However, there is a performance impact when this resizing is required, so you will want to ensure that your font and size will allow all text to be drawn within the defined space. This process is simplified if your image design allows sufficient space for all fields.

- As a part of this process, you will need to document the x/y coordinates and lengths of each of the fields for your new check format. The coordinates and lengths must be defined in inches. Most paint tools allow you to use the mouse and point at any pixel within the image and get the x/y coordinates. For example, GIMP will display these coordinates in either pixels, inches, or centimeters. This is a very detailed step but easily accomplished when you use a tool that provides the needed positional feedback.
- Once your modifications are tested, always backup your modifications to a zip file so you always have reference to these in case they are replaced during a future release installation.

If your custom check artwork is created by scanning your existing internal items, you may have color images that need to be converted to black and white before they can be used for Make / Generate. When this is the case, you will need to convert your images from color (RGB) to bi-tonal (black/white) images as required by standard image exchange. This can be done using various image tools. Here is an example of how to do this using GIMP:

- Load your color image.
- From the menu, select Colors / Components / Channel Mixer; check monochrome, preserve luminosity, and save.
- From the menu, select Colors / Brightness-Contrast, move the brightness slider to the left and move the contrast slider to the right. You may find that this is a repetitive process that you can continue to do until you achieve the best possible black / white version of your image. When OK when you are satisfied with the results.
- From the menu, select Image / Mode / Indexed; select use black and white (1 bit) palette and then use the convert button which will translate your image from color to black / white.
- From the menu, select File / Export As to write your transformed image to an external file. The image file name that you enter in the box at the top must have a valid image file extension which will determine the image format to be used. Our suggestion is to export as PNG since it is a lossless image format and is commonly used within the industry. You should not select TIFF since your exported image will not have valid TIFF tags per image exchange standards.

Options Editor

Tool Type	Tool Usage
System Editor	Editors maintain runtime tables which control the application environment.

The Options Editor allows you to view and modify program options. Our program options are designed to allow you to customize our desktop tools to best meet your needs.

X9Assist program options will take affect only when they are saved and the application is restarted. If you make option changes, the Options Editor will ask you if you want the changes saved and if you want the application restarted to allow the option changes to be established.

Options are stored in the / Documents / x9_assist / xml / options / folder. The system default options file is “programOptions.xml” which is loaded during startup and modified using the Options Editor.

You will note that the option panel is tabbed, allowing you to enter options that control a variety of internal functions. Options exist for the following areas:

- X9
- Validation
- File Locations
- GUI
- Colors
- Draft Check
- Settings

Saving an Options File

Use the Save button to save an options file after modifications have been made. You will need to restart for your modifications to become effective.

Resetting to Default Options

You can reset to system default options using the Reset button which is available on the tool bar. Reset will overwrite the options file with system defaults.

An options reset can also be accomplished by just deleting the “programOptions.xml” file (with folder / documents / x9_assist / xml / options /), forcing it to be automatically created in the next user session.

Help

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

The help facility provides information on various topics which can be viewed interactively as separate help topics.

Help documents are stored in HTML within the help\helpFiles\ folder. They can be browsed using our integrated help facility or can be launched using any internet browser.

Using our help index, you can double click any document and you will then be launched into a viewer for that particular help topic. You can open multiple documents at the same time.

All X9Assist documentation is available as a consolidated User Manual that can be downloaded from our website.

We would also appreciate your feedback on how our help related information can be improved. This includes better information on the topics that we have provided, as well as additional topics that are needed. Please contact us at x9assist@x9ware.com regarding suggested improvements.

Videos

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

We have created a series of videos that contain demonstrative help for all of our products, which includes X9Assist, X9Utilities, X9Ware-SDK, X9Ware-TiffKit, and X9Ware E13B-OCR. This content covers many of the areas where there are common questions, especially for the more advanced topics where we know this will be helpful.

These videos demonstrate a wide range of X9Assist related functions. FYI that you must license the appropriate product to perform certain X9Assist functions. For example, Make-Generate is only available with X9Assist. A given function will be grayed out on the X9Assist toolbar when it is not available per the current user license. A grayed out function indicated that it exists within the software, but is not available solely due to the user licensing level.

Our videos are hosted at https://www.x9ware.com/videos. We have stored these on our website with the intention to make them available to the largest possible user base. Our choice for this location is to eliminate potential problems within corporate environments, where your may be blocked from viewing videos that are stored on other sites such as YouTube or Vimeo. By doing this, we also eliminate ads, popups, and any potential problems associated with tracking or privacy.

We have several licensing levels of our desktop products: X9Vision, X9Validator, and X9Assist. They support the X9.37, ACH, and CPA005 file formats, which are licensed separately.

- X9Vision is our basic viewer, which displays records and fields with front/back images (for x9.37 files).
- X9Validator adds file/image validation, modify, repair, find, search/replace, export, merge, filtering, and IRD Print.
- X9Assist expands even further with make/generate (creating test files from spreadsheets), import, create, reversals, scrub (turning production files into test files by blocking personal account information), compare, repackage, and clone. These products are offered as 2-Seats, 5-Pack, 10-Pack, and Unlimited seats, all within your organization.

Our MP4 viewer makes every attempt to keep track of your unique positioning within the current video that is being watched. We hope this helps you to resume if your watch in interrupted and subsequently resumed.

Please feel free to let us know if there is additional content that would be helpful. We appreciate the viewing and thanks for watching!

About

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

The About panel provides more detailed information regarding X9Ware, our products, and recent enhancements. About is very helpful to identify the current release and build level. This information is needed when reporting problems to X9Ware (note that the release and build is also included in the system log, so this is not necessary when also sending a system log). The About panel also explicitly informs the user when a Candidate Build is being used. These are builds that still in development and need to be replaced with a “final” build when the release is completed by X9Ware.

There are scrollable panels on the left and right as follows:

- About includes a button which can launch a view a list of the third party software products (within the left side panel) that are used to build X9Assist and their associated license agreements. This list is always kept current by X9Ware as our third party tool usage might change from release to release.
- About includes the X9Ware user license agreement (within the left side panel) that was accepted as part of your application installation.
- About includes a list of the recent system enhancements (within the right side panel) that shows notes for the current and recent releases. Questions about any of these enhancement items should be forwarded to us.
- About has a “Help Index” button on the action line which will launch a viewer of all available help topics.

Registration

X9Vision ?	X9Validator ?	X9Assist?		X9.37 ?	ACH ?	CPA005 ?
YES	**YES**	**YES**		**YES**	**YES**	**YES**

The Registration panel is used to enter and subsequently review product authorizations.

Our issued license keys are in one of two formats.

- First is a 19-character license key that is provided through an email transmittal, as part of our invoicing and license key issuance process. These license keys are internally validated using a proprietary encoding scheme and will not require internet access.
- Second is a 32-character license key that is obtained from an online purchase. These license keys will be validated using an online registration process to our website, which will require internet access.

The two license key formats are readily identifiable due to their different layouts. You can visually inspect your specific license key and readily determine which format has been issued to you. The 19-character license keys are formatted as xxxx-xxxx-xxxx-xxxx with the embedded dashes. We have separate entry and validation procedures for these two formats.

The 19-character license key are referred to as “offline” license keys and are formatted as xxxx-xxxx-xxxx-xxxx). These keys are distributed within an email (transmittal) which is provided after your payment is either completed or confirmed as in progress. All fields (product type, customer name, company name, expiration date, and license key) are validated together as a group. After you enter the required information, you then use the register button (at the bottom of the panel) to initiate the registration process. An error in any of these fields will result in a mismatch and will thus require that you closely review and enter the information again (after the needed corrections are made). It is often easier to use copy and paste this information from your transmittal directly into the form, which will ensure that there are no typos or transcription errors.

The 32-character license keys are referred to as “online” license keys and will be registered through our website. These license keys are sent to you after an online product product from our website. As part of this registration, you will enter the product type and the license key as a functional pair, which must agree with your online purchase. After you enter the required information, you then use the register button (at the bottom of the panel) to initiate the registration process. It is often easier to copy and paste the license key into the form, which will ensure that there are no typos or transcription errors.

Online license keys will have to be manually entered as part of registration. Offline license keys do not need to be entered, because we distribute a copy of your license file that can be imported using the registration panel. If you have an off line key, then using import is much easier since it eliminates the potential for data entry errors.

If you must enter your license key information, please do so carefully. Note that you can cut and paste several of these fields (including the license key) from our transmittal which means that you do not have to re-enter this information. This can reduce the potential for input errors.

The following buttons appear on the main registration panel and are used to launch a tailored process for your product registration:

Button	Usage
Import license	Import a new license from an external file (for example, a new license that was provided to you as an email attachment). X9Ware LLC distributes these license files as part of offline purchases (not through our website). These are very small encrypted TXT files that include your specific licensing information. By importing this file, you eliminate the more tedious task of manually enter your license key. These license key files are distributed using a structured name which is formatted as: elicense_[company]_[product]_expires_[expirationDate].txt.
Enter 19-character license key	Enter a 19-character license key (these are formatted as xxxx-xxxx-xxxx-xxxx) which would be received via an email transmittal, either from X9Ware LLC or perhaps shared with you by someone else within your organization. These keys are associated with what we refer to as an offline purchase, which means that the purchase was through a direct invoicing process (the purchase was not made online from our website). This license key will be authorized locally and will not require internet access.
Enter 32-character license key	Enter a 32-character license key that you received from an online purchase or renewal via our website. You will be able to copy and paste the key (you will not have to re-enter it). Please be aware that the license key will be validated using a real-time confirmation to our website, so internet access is required to complete this registration process.
Export license	Export your current license key to an external file that you then share with others within your organization. This exported license can be distributed as an email attachment for subsequent import.

The following button actions are available at the bottom of the online and offline popup panels:

- Register – is used to initiate registration process.
- Clear – is used as a reset function that will clear any text fields that have been entered.
- Cancel – is used to exit the registration panel.

After the license key has been successfully entered or imported, an elicense.txt file is updated within folder / documents / x9_assist / license /. This file is updated only as part of this registration

process and will otherwise remain unchanged. The elicense.txt file is in an encrypted format, such that it cannot be viewed or modified. You can confirm that the elicense.txt file has been updated during the registration process using the associated date-time stamp that displayed by File Explorer.

Subject to your license level, the elicense.txt file can be shared with others within your organization. This is done by sending it to them (as an email attachment, etc) and having them drop it into the /documents/x9_assist/license/ folder on their respective workstation. This is then a replacement of their default or expiring license file, and will automatically bring their credentials current without having to know the license key or go through the license key entry process.

The elicense.txt file is unique to your company, so it should be handled accordingly and only shared within your organization, and then only within the limits of your license. For example, if you have a single seat license, you should not share your license with others, since they must obtain a unqiue and separate license for their workstation. You obviously need to ensure that your license is not shared outside of your organization. We have attempted to make our licensing process as painless as possible. Your cooperation and compliance is greatly appreciated.

Our licenses are issued for a duration of one or more years. You can purchase a multi-year license to eliminate expiration date issues. Our applications provide informational messages on any upcoming license expirations within sixty (60) days in advance of an upcoming expiration date.

Licensing requests can be made via email to x9assist@x9ware.com.