QAD Adaptive Applications
Enterprise Edition

Conversion Guide

QAD Enterprise Edition

78-0949-2022EE-Rev1
QAD Enterprise Edition 2022
March 2024

ProgressConversion_CG_v2022EE.pdf/ymg/hkr

QAD Inc.
100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
https://www.qad.com

Contents

QAD Enterprise Edition Conversion Change Summary

Product Name Changes

Starting in September 2019, the new name for QAD's complete portfolio of products is QAD Adaptive Applications. Additionally, QAD Adaptive ERP is the new name for QAD's flagship ERP solution. QAD Adaptive ERP includes the functionality previously associated with QAD Cloud ERP and QAD Enterprise Applications - Enterprise Edition, plus the QAD Enterprise Platform and Adaptive UX which resulted from the Channel Islands program. Going forward, the terms QAD Enterprise Applications, QAD Cloud ERP, and Channel Islands will be deprecated but will remain in previous documentation and training materials. QAD's intention is to—as soon as possible—eliminate the use of the deprecated terms going forward.

Change Summary

The following table summarizes significant differences between this document and the previous versions.

Date/Version	Description	Reference
March 2024/2022EE	Added a note about required parameters when running the conversion process.	page 77 and page 84
September 2022/2022EE	Updated software requirements to YAB 1.15 and Enterprise Edition 2022.	--
	Updated the *Rerun the Conversion Execution* section with a note.	page 53
March 2022/2021EE	Added details for executing conversions on environments that include Adaptive UX.	page 46 and page 77
September 2021/2021EE	Updated software requirements to YAB 1.13 and Enterprise Edition 2021.	--
September 2020/2020EE	Updated the package name required to convert a pre-EE environment to Enterprise Edition.	page 43
	Removed the section on manually removing SAQ packages. These steps are now handled automatically.	--
	Added information on how to successfully restore a source EE database into an environment that has enabled secure configuration.	page 76
	Updated the command required to update the environment scripts when upgrading an existing EE environment.	page 84
May 2020/2019EE-Rev1	Clarified that manual changes made to databases must be reapplied after a pre-EE to EE conversion is complete.	page 46
	Adding information on rerunning the conversion execution during an Enterprise Edition upgrade.	page 78

Date/Version	Description	Reference
September 2019/QAD 2019EE	Minor updates for the QAD 2019EE release and related conversion process.	--
	Added information for UTF-8 clients.	page 40
	If the source environment that is being upgraded includes SAQ, you must remove three SAQ packages before executing the conversion.	--
September 2018/2018EE	The conversion-ee-upgrade process now runs the operational and financial conversions as independent steps.	page 46
	Minor updates to Chapters 5 and 6.	page 73 and page 81
	The conversion-ee-upgrade process now uses the standard database structure and schema YAB processes, which dynamically calculate the changes to be applied.	page 84
	Updated Appendix H with conversion process validation.	page 203
December 2017/2017EE-Rev1	Added a note about user “mfg.”	page 41
	Updated Chapter 5 title and overview section, removed references to QDT.	page 73
	Added a note on Progress OpenEdge database conversion.	page 76
	Updated Chapter 6 title and overview section.	page 81
	Added a note about add-on products.	page 82
September 2017/2017EE	Updated package version numbers in Table 3.1.	page 43
	Removed prerequisites regarding SAQ upgrades.	---
	Introduced Chapter 6, Upgrading an Existing QAD Enterprise Edition Environment.	page 81
	Moved upgrade validation steps to Appendix H.	page 203
December 2016/2016EE-Rev3	Updated package version numbers in Table 3.1 and added a note to the yab-conv-pre-ee package name	page 43
	Updated install command in Conversion Prerequisites section	page 43
	Updated the Prerequisites section in Chapter 5 regarding SAQ upgrades	---
June 2016/2016EE-Rev2	Updated Table 3.1 with a Name column	page 43
	Updated package name for QAD 2016 Enterprise Edition Conversions Patch 1	page 43
	Added note to help users avoid lock table overflow errors	page 44
April 2016/2016EE-Rev1	Updated steps to convert from pre-EE to EE	page 38
	Added information on validating an upgrade execution	page 205
	Added Appendix F – AIM Conversion	page 193
	Added Appendix G – Snapshots	page 197

Date/Version	Description	Reference
March 2016/2016 EE	Changed name of book to *Enterprise Edition Conversion Guide*	---
	Introduced using the YAB Installer to convert systems	---
	Remove references to QDT	---
	Major revisions to Ch. 3 Conversion Execution	page 35
	Major revisions to Ch. 5 Upgrading EE	page 73
	Multiple revisions to Appendix D – Conversion Troubleshooting	page 183
	Updated Appendix E – Log Files with YAB information	page 189
	Removed Log File Naming Conventions from Appendix E – Log Files	---
	Removed Appendix – Configuring Access to Progress Editor	---
	Removed Appendix – Snapshots	---
	Removed Appendix – Configuring QDT to Use New Progress or Tomcat Servers	---
October 2015/2015 EE-rev2	Added SAQ requirements information to Prepare Environment	---
September 2015/2015 EE-rev1	Various editorial changes	---
	Revised the Reporting Currency parameter description	page 32
	Revised the Preserve Custom System Data section	---
March 2015/2015 EE	Various editorial changes	---
	Revised Red Hat QDT start script information	---, page 162
	Documented a functionality change that affects the selection of components during upgrades	---
	Updated figure to show that the components available for upgrade are deselected by default	------
	Updated figure to document the new DB Connections field in the Compiler Settings screen	---
	Removed the section Upgrading Only the .NET UI Component	page 82
September 2014/2014 EE-Rev 1	Various editorial changes	---
	Reorganized and revised the Install and Configure Software section of Upgrading QAD Enterprise Edition	---
	Documented a limitation that prevents upgrading only the .NET UI component in 2013.1 EE to 2014 EE	page 82
March 2014/2014 EE	Various editorial changes	---
	Revised Tax ID Utility (uxtaxid.p) description	page 20
	Revised Default Daybook Sequence Effective Date (MM/DD/YYYY) description	page 29
	Revised Create Live Main Database	---
	Added entry for Update Contract Revenue Account (utsarrv.p)	page 57
	Updated the Upgrading QAD Enterprise Edition chapter	page 73
	Updated Preserving Custom System Data	---
	Updated Pause Points - convprogpauselist.ini section	---
	Updated Creating a Snapshot section	---
	Updated the Configuring QDT to Use a New Progress or Tomcat Version appendix	page 161
September 2013/2013.1 EE-rev 1	Added section on Upgrading Only the .NET UI Component	page 82

Date/Version	Description	Reference
September 2013/2013.1 EE	Extensive editorial changes	---
	Added Preserve Custom System Data section to Conversion Execution chapter	page 40
	Added information about financial attachment considerations when moving a database	page 74
March 2013/2013 EE	Extensive editorial changes	---
	Removed the section Upgrading Only the .NET UI	---
	Reorganized Upgrading QAD Enterprise Edition chapter	page 73
September 2012/2012.1 EE	Changed Management Currency entry in table to Reporting Currency	page 29
March 2012/2012 EE	Added section: Backup Progress Property Files	---
	Added text on what to do in the event of premature conversion termination due to data-related errors.	---
	Added information regarding how upgrading affects process map configuration data	---
	Documented additional tasks required when upgrading the QAD UI only from QAD EE 2011.1 to QAD EE 2012.	page 73
	Removed and replaced text in Conversion Accounts section	page 157
	Added section: Performing a Clean QDT and Enterprise Edition Reinstallation	---
	Added appendix: Configuring QDT to Use a New Progress Version	page 161
September 2011/2011.1 EE	Documented Red Hat 6 environment considerations	---, page 162
	Documented the impact of database management using conmgr.properties and Progress DBMAN tools on upgrades	---
	Added appendix: Configuring QDT to Use a New Progress Version	page 161

Chapter 1

Conversion Overview

This chapter describes the requirements and process for converting a Progress database to QAD Enterprise Edition.

Introduction

This chapter provides an overview of the conversion process and describes the skills required to perform a database conversion.

The QAD YAB Installer provides processes that can perform:

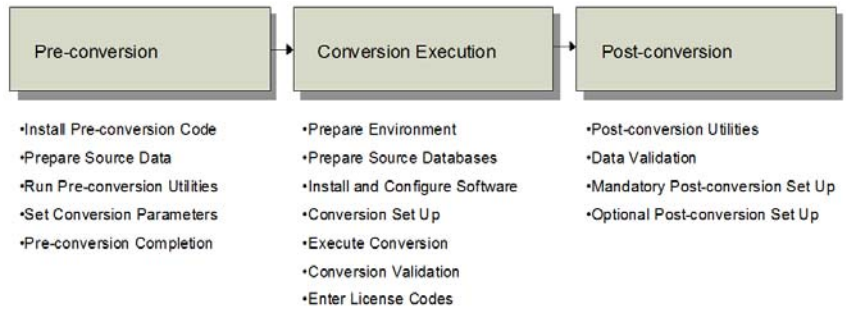
- Pre-EE to EE conversion
- EE to EE upgrade

QAD Enterprise Edition Conversions

The QAD Enterprise Edition conversion process covers the steps for converting the database of an earlier QAD ERP system version to QAD Enterprise Edition. Figure 1.1 summarizes this process.

Note This overview is for the pre-EE to EE conversion option only. It does not apply to the EE to EE upgrade option.

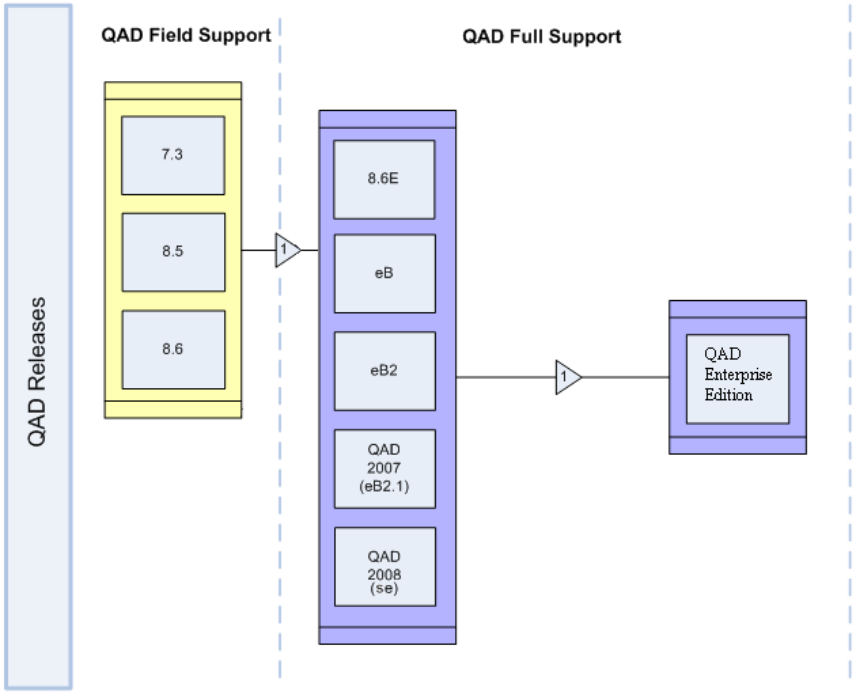
Fig. 1.1
Conversion Summary



Conversion Path Diagram

Figure 1.2 illustrates the QAD Enterprise Edition conversion path. As depicted in the diagram, direct conversions to QAD Enterprise Edition are supported for MFG/PRO versions 8.6E and above only.

Fig. 1.2
QAD Enterprise Edition Conversion Path



If the database to convert uses a version before 8.6E, update it to a supported version (8.6E and above) before conversion to QAD Enterprise Edition.

Prerequisite Skills

The procedures in this manual are only intended for system administrators and other qualified personnel with the following training and experience:

- QAD Enterprise Edition installation and administration training
- A working knowledge of QAD Enterprise Edition products
- System maintenance expertise (for example, adding menu items and custom programs)
- Starting and stopping AdminServers, AppServers, and NameServers
- Basic Progress database administration skills (`probkup`, `prorest`, and dump and load)
- Familiarity with Linux/UNIX and command-line navigation

QAD strongly recommends that all people involved in the conversion have the following background:

- Progress database tuning parameters
- Basic Progress coding skills for debugging simple Progress queries
- Basic Tomcat administration skills (starting, stopping, and installing Tomcat and changing ports and permissions)

Chapter 2

Pre-conversion

This chapter describes the preparation of a database for conversion to QAD Enterprise Edition.

Introduction

This chapter describes the pre-conversion steps required to prepare a database for conversion to QAD Enterprise Edition (QAD EE). It covers pre-conversion code installation, data preparation, conversion preparation, and pre-conversion finalization.

Note This chapter only applies when converting from versions before Enterprise Edition. It is not required when upgrading an existing Enterprise Edition installation.

Pre-conversion Overview

The pre-conversion process prepares the source database for conversion by correcting the following types of issues:

- Data that is missing from the old Financials, but required in the QAD Enterprise Edition Financials (for example, Entity on employee records)
- Data that exists in the old Financials, but is not allowed in the same state in QAD Enterprise Edition Financials (for example, blank credit terms code on customers and suppliers)
- Data that is invalid in both the old and new Financials (for example, VAT registration IDs that do not conform with the European Commission's specified format for the applicable country)

The Pre-conversion process consists of the following major tasks:

- Install Pre-conversion Code
- Prepare Source Data
- Run Pre-conversion Utilities
- Set Conversion Parameters
- Pre-conversion Completion

The database is prepared using pre-conversion utilities and reports to highlight items to change or update for the conversion. These run in a character environment only. All text is in English and no translations are available for these reports and utilities.

The pre-conversion utilities and reports do not appear on any menus. They are executed by typing the program name at any menu command line. The programs do not support CIM and cannot run in batch mode.

Install Pre-Conversion Code

The EE media contains all of the pre-conversion reports and utilities required to prepare a database for conversion. The pre-conversion source code is version independent and supports MFG/PRO versions 8.6E through QAD Standard Edition and Progress versions 8.3e to the latest Progress version. As a result, attempting to compile the pre-conversion code causes Progress errors due to field and table references that do not exist in all versions.

Therefore, QAD recommends running these programs as uncompiled source code.

To install the pre-conversion code, use the following steps:

1. The pre-conversion code is located in the `etc/preconvrep/preconvrep.zip` file.

   The `preconvrep.zip` file is included on the 2022EE image. The `.zip` file is on the 2022EE image but it is not installed because it is designed to be installed on a pre-EE environment (SE or eb2.1).

2. Unzip the pre-conversion code into its own directory. QAD recommends using the `preconvrep` directory.

3. Add the following directories to the beginning of the `PROPATH`:
   - *<preconvrep_install_directory>*
   - *<preconvrep_install_directory>*`/us`

You can then run each program by typing the name at the character menu command line.

Conversions from eB2.1 or Later

If you are converting from eB2.1 or later, the `preconvrep.zip` file contains data (*.d) files that you must load into the qadadm and main databases for the Domain Table Extension utilities. The following table describes into which database the files are loaded.

Table 2.1
Database File Inventory

Database	Files in <*preconvrep_install_directory*>
qaddb	`cd_det.d`, `lpmd_det.d`, `mnd_det.d`, `mnt_det.d`
qadadm	`lbl_mstr.d`, `pgmi_mst.d`

If you are running eB2.1 SP6 (QAD 2007.1) or later, you receive errors when loading some of these files because the system already contains this data. You can ignore these errors.

These data files contain trailer details with settings related to the database from which they were dumped. The following figure shows one such trailer.

Fig. 2.1
System Data File Trailer Section

```
.
PSC
filename=mnd_det
records=00000002
ldbname=qaddb
timestamp=2007/08/28-00:03:01
numformat=44,46
dateformat=mdy-1920
map=NO-MAP
cpstream=ISO8859-1
.
```

These settings may not match your settings. If they differ, temporarily change your database startup parameters to match QAD's. You can also edit the trailer details of each file to reflect your database startup parameter settings to ensure that the data loads properly.

The following table lists the values that QAD uses for its startup options.

Table 2.2
Startup Option Values

Option	Parameter Name	QAD Parameter Setting
`-d`	Date Format	mdy
`-yy`	Century Year Offset	1920
`-ld`	Logical Database Name	qaddb and qadadm
`-numdec`	Numeric Fractional Separator	44,46 (comma for thousands, period for cents)
`-cpstream`	Stream Code Page	ISO8859-1
`protermcap-entry`	`PROTERMCAP` file setting to indicate whether to use a different character translation from the current stream when reading an input stream.	NO-MAP

In addition, you must compile the following programs and put them into the `us/ut` subdirectory under the directory where the pre-conversion code was installed:

- `utqtabs.p`
- `utqtabsa.p`
- `utqtplsd.p`
- `utqtpsda`

Note If you are running eB2.1 SP6 (QAD 2007.1) or later, you already have `utqtabs.p`, `utqtabsa.p`, `utqtplsd.p`, and `utqtpsda` installed and do not need to compile these files.

Prepare Source Data

Table Extension Domain Utilities (eB2.1 and later only)

For conversions from eB2.1 and later, you must update selected QAD Table Extension (`qtbl_ext`) records with domain information before running the conversion. This update is done using the following utilities:

- `utqtabs.p` – Covers supplier lots on shippers
- `utqtplsd.p` – Covers Expense Due account in Sales Account Maintenance (1.2.17, `ppplsmt.p`)

Run these utilities in update mode once from any domain even if these features are not being used (all domains are updated in a single execution of each utility). They provide a simulation mode for previewing updates.

The impact of these utilities on future transaction processing depends on how extensively their associated features are used. If supplier lots in shippers are used, the supplier lot is obtained from the correct domain.

Archive/Delete Transaction Data (optional)

You can reduce the time required to complete the conversion process by archiving and deleting as many records as possible before conversion. This action is optional, but recommended.

You can use the following functions to archive and delete data:

- GL Transaction Consolidation (25.13.11, `glcons.p`)

  Transaction consolidation should include all accounts and be done for single accounting periods per consolidation.

  Warning Do the consolidation by period, not account, and preferably by month as opposed to by year. For consolidation by year, any given account has an opening balance but zero period amount for any period other than the period for the consolidation effective date.
- Invoice History Delete/Archive (7.13.23, `soivup.p`)
- Uninvoiced Receipt Delete/Archive (28.22, `aprcup.p`) *
- Uninvoiced Logistics Charge Delete/Archive (28.21, `aplaup.p`)
- Closed PO Receipt Delete/Archive (5.22, `porcup.p`) *
- Closed PO Delete/Archive (5.23, `popoup.p`) *
- Supplier Schedule Delete/Archive (5.5.3.23, `rsdel.p`) *
- PO Shipper Delete/Archive (5.13.23, `rsscdel.p`) *
- Customer Schedule Delete/Archive (7.5.23, `rcdel.p`) *
- PRO/PLUS Sequence Schedule Detail Delete (7.5.4.22, `rcsqscdl.p`)
- Shipper Delete/Archive (7.9.23, `rcsdel.p`) *
- Container Delete/Archive (7.7.23, `rcctdel.p`) *
- Expired Quote Delete/Archive (7.12.23, `sqqoup.p`)
- Inventory Transaction Delete/Archive (3.21.23, `ictrup.p`) *

- Zero Balance Delete/Archive (3.23, `icldup.p`)
- Cost Set Delete (30.23, `cscsdel.p`)
- Intrastat Delete/Archive (2.22.23, `iehup.p`)
- Retired Asset Delete/Archive (32.23, `fartup.p`)
- Tag Delete/Archive (3.16.23, `pitdup.p`)
- WIP Lot Trace Delete/Archive (3.22.13.23, `wldel.p`) – PRO/PLUS Module
- Intersite Request Delete/Archive (12.15.23, `dsdmup.p`)
- Closed Intersite Demand Delete/Archive (12.17.23, `dsdoup.p`)
- Product Structure Delete/Archive (13.23, `bmpsdel.p`)
- Routing Delete/Archive (14.13.23, `rwrodel.p`)
- Work Order Delete/Archive (16.23, `wowoup.p`)
- Operation History Delete/Archive (17.23, `sfopup.p`)
- Cumulative Work Order Delete/Archive (18.23.2, `recwoup.p`)
- PCR/PCO Delete/Archive (1.9.15, `ecarcdel.p`) – PCC Module
- Lot Master Delete/Archive (1.22.23, `clltup.p`) – Compliance Module
- Service/Repair Order Delete/Archive (7.23.23, `srsroup.p`)
- Closed Project Delete/Archive (10.23.23, `pjpjup.p`) – PRM Module
- Call/Quote History Delete/Archive (11.1.1.23, `fscaarc.p`)
- Service Request Delete/Archive (11.1.15.23, `fssrarc.p`)
- Field Notification Delete/Archive (11.3.12.23, `fsfnarc.p`)
- Contract Delete/Archive (11.5.13.23, `fssaarc.p`)
- Revenue Delete/Archive (11.5.18.23, `fsdefarc.p`)
- Flow Delete/Archive (17.21.23, `flschup.p`)
- QM Test Results Delete/Archive (19.26.22, `mpcaup.p`)
- QM Quality Order Delete/Archive (19.26.23, `qcqcup.p`)
- Family Hierarchy Delete/Archive (33.3.23, `spfhup.p`)
- Operations Plan Delete/Archive (33.15.23, `spfpup.p`)
- Simulation Delete/Archive (33.17.23, `spfspup.p`)

* These activities provide the greatest benefit.

To archive or delete data, start the desired utility by entering the utility name (or menu number) on the command line. After you have completed any deletion or archiving, skip to “Global Tax Management Conversion (pre-eB)” on page 12.

Global Tax Management Conversion (pre-eB)

If you are converting from MFG/PRO eB or later, skip to the next section. If you are converting from a release before MFG/PRO eB, convert the tax environment to use Global Tax Management.

See Appendix A, “GTM Conversions,” on page 85 for more information.

Run Pre-conversion Utilities

Many changes to existing static data are required before conversion to QAD Enterprise Edition. Some of the utilities described in this section provide the means to perform mass updates, while other utilities are required to create the underlying data used in the conversion.

All of the utilities and reports listed in this section prompt you to specify a report output directory. Once supplied, this location is saved and used as the user default for each subsequent execution of a utility or report. You can change this value and the new value is saved as the new output directory for that user (for example, each user can have a separate output directory location).

Note The Data Preparation Report is described in more detail in a later section and is the pre-conversion master report. It highlights all data errors and the recommended utilities to correct them. It also highlights which utilities you must run before beginning a conversion. The first time this report is executed, it can produce many errors. To minimize the number of errors in the Data Preparation Report, QAD recommends running the following utilities before the Data Preparation Report:

- Employee Entity (`uxempent.p`)
- End User Contact (`uxendupd.p`)
- Control Account Utility (`uxctrl.p`)
- GL Account/Project Range (`uxglproj.p`)
- Orphaned Address (`uxaddrfix.p`)
- Credit Terms (`uxcrterms.p`)

Employee Entity Utility (uxempent.p)

QAD Enterprise Edition requires every employee in Employee Master (emp_mstr) to have an entity code. This utility reports and updates employees with a user-specified entity code.

- It shows employees missing an entity code and employees already assigned an entity code.
- To see employees missing entity codes, leave the Entity field blank and Update = No.

You can run the utility for a range of employees when you must assign multiple entity codes.

The utility provides a simulation mode that allows you to preview the effects of the update.

Run this utility. For eB2.1 and later, run this utility for each active domain.

The report name is `uxempent-<dbname>-<domain>-<date>_<time>.prn`.

This utility does not affect future transactions; it updates an unused QAD reserved field in the emp_mstr table.

End User Contact Utility (uxendupd.p)

QAD Enterprise Edition requires every end user (eu_addr) to have one primary contact with a unique contact name. This utility reports and updates end users who are missing a contact name, have duplicate contact names, or have multiple primary contacts.

The utility provides a simulation mode that allows you to preview the effects of the update.

The report name is `uxendupd-<dbname>-<date>_<time>.prn`.

For eB2.1 and later, one execution processes all active domains.

Correct some of these errors through End User Maintenance (11.9.1, `adeumt.p`).

This utility can impact future processing by causing the contact names and primary contact designations to differ from the information in previous transactions.

Control Account Utility (uxctrl.p)

The conversion routines must know which accounts are used as Accounts Receivable and Accounts Payable control accounts in the AR and AP modules of the pre-conversion installation.

The conversion cannot determine this information because the routines overlook accounts not referenced to customers, suppliers, or transactions.

The utility prompts you for lists of account codes currently used as control accounts. If it is current practice, you can specify the same account for AR and AP. However, the GL Account Type Utility will ultimately indicate that one of these accounts must be different after conversion. See *GL Account Type Utility (uxglacup.p)* on page 23 for more information.

The report output file name is `uxctrl-<dbname>-<domain>-<date>_<time>.prn`

Run this utility as many times as necessary. Previous answers are displayed and can be removed if desired. For eB2.1 and later, it is run in each active domain.

The GL Account Type Utility, Data Preparation Report, and the conversion later use the inputs to this utility.

This utility determines the GL account codes to define as customer and supplier control accounts within Enterprise Edition. It does not affect the current system or transactions. It only creates QAD Work Table (qad_wkfl) records that store the user's input values.

Note If the source database does not use the Accounts Payable and/or Accounts Receivable modules, and the Chart of Accounts does not contain default General Ledger account codes for Customer and/or Supplier Control Accounts, create new General Ledger accounts. Reference them within the utility for each domain if the source database version is eB2.1 or above. You also reference these account codes elsewhere within the database. However, the output from other pre-conversion utilities directs the user as necessary.

GL Account/Project Range Utility (uxglproj.p) if pre-eB

The following information only applies if you are an MFG/PRO 8.6E or 9.0 user. If you are not, go to the next section.

Before version eB, project codes were not a validated part of the GL account structure. When project codes were in use, they could be used with all valid GL account, sub-account, and cost center combinations.

The conversion must know which GL accounts are valid for use with project codes in the QAD Enterprise Edition Financials. This information allows the conversion to correctly set the analysis settings of the GL Accounts (that is, Analysis Type and Analysis Limitation).

Note Where a GL account is determined to allow project codes, Chart of Account (COA) Mask records are created. The records allow you to use projects with all valid combinations of sub-account and cost center for that account. Review the COA Mask records after the conversion finishes.

This utility prompts you for GL account code ranges to use with project codes after conversion. Note, however, that not all GL accounts are allowed to use project codes after conversion.

The report output file name is `uxglproj-<dbname>-<date>_<time>.prn`.

Run this utility as many times as necessary. Previous results are displayed and can be removed if desired.

This utility does not affect future transactions; it only creates QAD Work Table (qad_wkfl) records storing the user's input values.

Orphaned Address Utility (uxaddrfix.p)

To be converted, all customer, supplier, end user, and remit-to addresses must have a related Address List Detail (ls_mstr). The Data Preparation Report indicates if you must run this utility.

This utility reports and creates missing Address List Details (ls_mstr) in the following scenarios:

- A customer exists with Customer Master (cm_mstr) and Address Master (ad_mstr) but no ls_mstr record for type "customer."
- A supplier exists with Supplier Master (vd_mstr) and ad_mstr but no ls_mstr record for type "supplier."
- An end user exists with Service/Support End User Master (eu_mstr) and ad_mstr but no ls_mstr record for type "enduser."
- A supplier exists and has a reference to a remit-to address (vd_remit <> " ") that has a valid address but no ls_mstr record for type "remit-to."
- A site exists with an address, but no ls_mstr of type "company" is found.

The utility provides a simulation mode that allows you to preview the effects of the update.

For eB2.1 and later, one execution processes all active domains.

The report output file name is `uxaddrfix-<dbname>-<date>_<time>.prn`.

This utility can affect the system by making previously unusable addresses usable.

Credit Terms Utility (uxcrterms.p)

All customers and suppliers must have a valid non-blank credit terms code assigned to them before converting to QAD Enterprise Edition. The Data Preparation Report indicates if you must run this utility.

A blank credit terms code (that is, ct_code = <blank>) is no longer permissible and the conversion deletes it. To simulate the scenario of a customer or supplier with no credit terms, create a credit terms code in Credit Terms Maintenance (2.19.1, `adcrtmt.p`) with 0 Disc Days and 0% Disc Pct.

Any credit terms code assigned to customers or suppliers must exist in Credit Terms Master (ct_mstr).

Run the utility separately for customers and suppliers. You can run the utility for a range of customers or suppliers when you must assign multiple credit terms codes. It reports any customers and suppliers having a blank or invalid credit terms code and updates missing terms with the user-specified code.

To see customers or suppliers with invalid credit terms assigned, leave the Credit Terms field blank and set Update to No.

The utility provides a simulation mode that allows you to preview the effects of the update.

The report output file name is `uxcrterms-cust-<dbname>-<domain>-<date>_<time>.prn` or `uxcrterms-supp-<dbname>-<domain>-<date>_<time>.prn`.

For eB2.1 and later, run the utility for each active domain.

This utility alters future transactions involving the customers or suppliers updated with new credit terms.

Data Preparation Report (gpdatarp.p)

This report highlights any data issues you must resolve before conversion such as:

- Data that is missing from the old Financials, but required in the new Financials.
- Data that exists in the old Financials, but is not allowed in the same state in the new Financials.
- Data that is invalid in the old and new Financials.

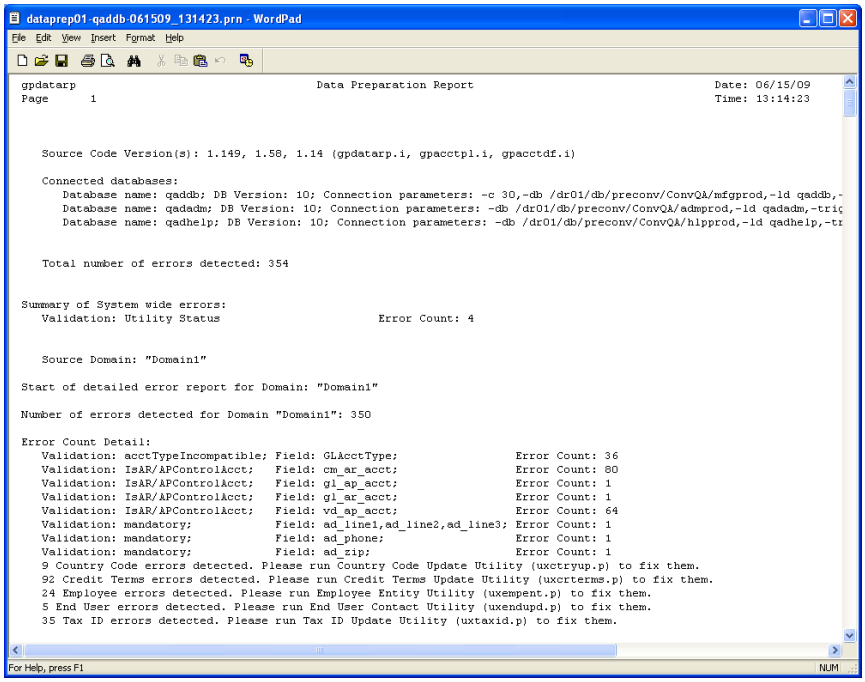
Warning If this report does not indicate an error, review the county and state data in eB2.1 and later before conversion to eliminate similar or inconsistent entries. Compare pre- and post-conversion state and county data to verify data integrity. Otherwise, the conversion could make state and county data unusable.

This report is the gatekeeper for the conversion. You cannot run the conversion until this report has zero errors.

Although the name indicates that it is a report, it creates a QAD Work Table (qad_wkfl) record in the database once no errors are found. The conversion looks for this record before it begins.

The report output file name is `dataprep#-<dbname>-<date>_<time>.prn`. Where # represents a segment number. The report can have multiple segments.

Fig. 2.2
Data Preparation Report



For eB2.1 and later, one execution processes all active domains.

The following new utilities are provided to assist with mass correction of some data errors:

- Country Code, (`uxctryup.p`)
- Tax ID (`uxtaxid.p`)

Run these utilities if the Data Preparation Report indicates that there are errors in these areas. Their use is optional because the data can alternatively be manually corrected through the menu functions listed on the respective utility's reports.

You probably must run the Data Preparation Report multiple times before all the errors are resolved.

Running this report does not affect future processing.

Account Type Incompatibility Error

Most errors shown on the Data Preparation Report are self-explanatory. One exception is the Account Type Incompatibility error. In the old Financials, you can define an account in Account Code Maintenance as an Expense type account. You can use that account code in static data for non-expense type purposes (for example, in Product Line Maintenance as an Inventory account).

The QAD Enterprise Edition Financials require that accounts used for specific purposes must have the appropriate account type definition based on the financial statement where they appear. You define account codes used in database fields for static data associated with the balance sheet as Type (ac_type) A(sset) or L(iability); not I(ncome) or E(xpense) in Account Code Maintenance (25.3.13, `glacmt.p`).

Similarly, you define account codes used in database fields for static data that are associated with the income statement as type I(ncome) or E(xpense); not A(sset) or L(iability).

For example, the Inventory Account Code (pl_inv_acct) or the WIP Control Account (pl_wip_acct) used in Product Line Maintenance is defined as an Asset or Liability type account, preferably Asset. The important thing is that the account code is represented on the correct financial statement. It is not acceptable to use an account code normally used for Purchases Expensed as Inventory or WIP Control Accounts.

There are two ways to resolve Account Type Incompatibility errors:

- If the account code also appears as a conflict on the GL Account Type Utility report, you can correct the account code. Run the utility in update mode and specify an alternate account code to resolve the conflict. The replacement account is applied during conversion and therefore has no affect on transactions occurring before conversion.
- If the account code does not appear as a conflict on the GL Account Type Utility report, the only course of action is to update the field in error (for example, pl_inv_acct). The field is updated through its maintenance program using an account code having the appropriate account type.

Warning

Using the second method to resolve Account Type Incompatibility errors affects future transactions once you make such a change. Therefore, it is only done after closing the database to transaction processing in preparation for conversion.

Do not correct these errors by writing a Progress utility to update the account type on the offending accounts. Doing so causes the Balance Sheet to be out of balance and Retained Earnings to no longer match the amount initially recorded.

Changes to correct these errors will result in information being reported differently on financial reports after conversion. Continuing the example from before, if a Purchases Expensed account were used as the Inventory account in Product Line Maintenance, any transactions involving this account appear on the Balance Sheet post-conversion. Before, they were on the income statement. For instance, a PO receipt involving such an account is reflected as inventory on the post-conversion balance sheet.

The Account Type Incompatibility errors are a result of the tighter controls used in the QAD Enterprise Edition Financials.

Beginning with QAD 2010.1 Enterprise Edition, Closing accounts are no longer restricted to Asset or Liability type accounts. Any account type (A, L, I, E) is allowed.

Another change introduced in QAD 2010.1 Enterprise Edition is that the Rounding Differences account is no longer restricted to an Income or Expense type account. Any account type (A, L, I, E) is permitted. The Purchase Order Receipts account now must be a Liability type account.

The impact of these changes may be apparent in the Account Type Incompatibility section of the Data Preparation Report. The changes may also be visible when choosing replacement accounts in the GL Account Type Utility for these types of accounts. The changes are apparent to the extent that exceptions involving these account types exist in the pre-conversion environment.

Table 2.3 lists the expected category for each Enterprise Edition account type.

Table 2.3 Expected Category by Account Type

Account Type	Category
Bank Account	Asset or Liability
Closing Account	any
Cross-Company Control Account	Asset or Liability
Customer Control Account	Asset or Liability
Customer Payment Account	Asset or Liability
Fixed Assets Account	Asset or Liability
Inventory Control Account	Asset or Liability
Supplier Control Account	Asset or Liability
Supplier Payment Account	Asset or Liability
Tax Account	any
WIP Control Account	Asset or Liability
Purchase Order Receipts	Liability
Realized Exchange Gain	any
Realized Exchange Loss	any
Result of Previous Years	Asset or Liability
Result of the Current Year	Asset or Liability
Rounding Differences	any
Unmatched Invoices	Asset or Liability
Unrealized Exchange Gain	any
Unrealized Exchange Loss	any

Unposted GL Transactions Error

The Data Preparation Report checks for any unposted GL transactions by accumulating General Ledger Unposted Transaction Detail (glt_det) records. It is possible that some of these records can be for a zero transaction amount and therefore do not appear on the Unposted Transactions Register (25.13.14, `glutrrp.p`). You can delete these zero-amount records using GL Transaction Delete/Archive (36.23.2, `mgmgrp01.p`).

Country Code Utility (uxctryup.p)

QAD Enterprise Edition requires that all addresses and employees have a non-blank, valid country code.

This utility reports addresses and employees having a blank or invalid country code. When executed in update mode, the utility updates these records with a user-specified country code. If execution of this utility is necessary, it is noted on the Data Preparation Report.

You can specify a range of addresses (which also covers employees) when multiple country codes must be assigned.

The utility provides a simulation mode that allows you to preview the effects of the update.

To see addresses and employees with invalid country codes, leave Country Code blank and set Update to No.

The report output file name is `uxctryup-<dbname>-<domain>-<date>_<time>.prn`.

Correct any discrepancies identified in the report by running the utility in update mode.

For eB2.1 and later, run it for each active domain.

This utility will potentially alter future transactions involving tax calculations due to the changed country code on customer and/or supplier addresses updated with new country codes. For this reason, it can be preferable to delay running this utility until the database is closed to transaction processing.

Tax ID Utility (uxtaxid.p)

QAD Enterprise Edition requires all US suppliers for 1099 reporting (vd_1099 = true) to have unique non-blank Federal tax IDs (ad_gst_id).

This utility reports duplicate or missing Federal tax IDs and lists VAT registration IDs that do not comply with the ISO format, when applicable. If execution of this utility is necessary, it is noted on the Data Preparation Report.

Additionally, if Validate VAT Registration in Global Tax Management Control (2.13.24, `txtxcmt.p`) is true, VAT registration IDs for customers and suppliers in EU countries must conform to their country's ISO format.

If Validate VAT Registration in Global Tax Management (2.13.24 - txtxcmt.p) is later disabled (for all Domains) to eliminate VAT registration ID errors on the Data Preparation Report, the conversion does not create the validations for VAT registration IDs. If users desire such validations post-conversion, they must manually create them or install a QAD-provided XML file containing predefined validations post-conversion. They then make any necessary corrections to VAT registration IDs that fail the validations.

For eB2.1 and later, run the utility in each active domain. If the Validate VAT Registration is true for any single domain, the validation is considered true for all domains. This action is necessary because Business Relations are created from the address records and Business Relations are system-wide within QAD Enterprise Edition.

The utility provides a simulation mode that allows you to preview the effects of the update.

The report output file name is `uxtaxid-<dbname>-<domain>-<date>_<time>.prn`.

Correct any discrepancies identified in the report by running the utility in update mode. If 1099 reporting is required, manually correct these errors through Supplier Maintenance (2.3.1, `advnmt.p`).

If 1099 reporting is not used, set this field to No in the utility and it assigns vd_1099 = false for all U.S. suppliers, reducing the number of Federal tax ID errors.

Correct VAT ID errors (if applicable) through Supplier Maintenance and/or Customer Maintenance (2.1.1, `adcsmt.p`) and/or Company Address Maintenance (2.12, `admgmt06.p`), based on the values in the List column of the report. Invalid VAT IDs are listed in the State ID column of the report.

The utility only affects 1099 reporting for US-based users and only when producing year-end 1099 forms.

Code Search and Replace Reports (eB2.1 and later only)

Run this suite of reports/utilities only if you are converting from eB2.1 and later. If you are not converting from eB2.1 and later, go to “Set Conversion Parameters” on page 29.

Use of these utilities and reports is optional (unless indicated in the Data Preparation Report), but recommended. They require that you run Table Extension Domain Update Utilities first.

The Code Search and Replace Reports are used to highlight conflicts in several areas that changed between eB2.1 and QAD Enterprise Edition, QAD 2007 and Enterprise Edition, and Standard Edition and Enterprise Edition.

Due to schema and architecture changes introduced in QAD Enterprise Edition, the scope of some data objects has changed from domain level to system level.

During the conversion, if two records exist with the same code in more than one domain, they are assumed to be the same and only one record is converted. As a result, some domain-level data could potentially be lost when converting a database with more than one domain.

The purpose of the Code Search and Replace Utilities is to provide a way to identify and correct instances of duplicate domain-level data.

Examples of Data Moved from Domain Level to System Level

- In QAD Enterprise Edition, addresses remain at the domain level, but map to business relations, which are at the system level in the new Financials. Each business relation must have a unique code.
- In QAD Enterprise Edition, entities remain at the domain level, but map to companies (now called entities) in the new Financials. The entity code must be unique in the database going forward.
- Before QAD Enterprise Edition, credit ratings were stored in the Generalized Code Master (code_mstr) table against field cm_cr_rating. Generalized Code Master is a domain-level table in eB2.1 and later. In QAD Enterprise Edition, credit ratings are stored in a new Financials table and become system-level data.
- Before QAD Enterprise Edition, customer and supplier types were stored in code_mstr against fields cm_type and vd_type. In QAD Enterprise Edition, customer and supplier types are stored in a new Financials table and become system-level data.
- Before QAD Enterprise Edition, voucher types were stored in code_mstr against field vo_type. In QAD Enterprise Edition, voucher types are called purchase types and stored in a new Financials table as system-level data.
- State and county codes are stored in the code_mstr against fields ad_state and ad_county. In QAD Enterprise Edition, states and counties become system-level data.
- Before QAD Enterprise Edition, tax classes were stored in code_mstr against field taxc_taxc. In QAD Enterprise Edition, tax classes are stored in a new table Tax Class Master (txcl_mstr) and become system-level data.
- Before QAD Enterprise Edition, tax usages were stored in code_mstr against field tx2_tax_usage. In QAD Enterprise Edition, tax usages are stored in the new table Tax Usage Master (txu_mstr) and became system-level data.

- Before QAD Enterprise Edition, tax types were stored in code_mstr against field txt_tax_type. In QAD Enterprise Edition, tax types are stored in a new table Tax Type Master (txty_mstr) and become system-level data.
- Tax zones are stored in the Tax Zone Master (txz_mstr) table. This table is a domain-level table in eB2.1 and later. In QAD Enterprise Edition, the domain field was removed and tax zones are system-level data.
- Tax environments are stored in the Tax Environment Master (txe_mstr) and Tax Environment Detail (txed_det) tables. In QAD Enterprise Edition, tax environments are system-level data. Before QAD Enterprise Edition, they were domain-level data.
- Credit terms are stored in the Credit Terms Master (ct_mstr) and Credit Terms Detail (ctd_det) tables. In QAD Enterprise Edition, credit terms are system-level data. Before QAD Enterprise Edition, they were domain-level data.
- Rounding methods are stored in the Rounding Method Master (rnd_mstr) table. Before QAD Enterprise Edition, they were domain-level data. In QAD Enterprise Edition, rounding methods are system-level data.

Duplicate Code Report (utsarrp1.p)

This report highlights codes (entities, credit ratings, credit terms, tax parameters, and so on) occurring in multiple domains. These codes were moved to the system level in QAD Enterprise Edition, and only one instance of each code can remain after conversion. This report allows you to identify any differences in the codes across domains. One execution spans all active domains.

The report output file name is `utsarrp1-dtl-<dbname>-<date>_<time>.prn` or `utsarrp1-sum-<dbname>-<domain>-<date>_<time>.prn`, depending on the report option selected.

Use the Replace Code Utility (`utsarrp3.p`) to correct any issues.

Code Usage Report (utsarrp2.p)

This report highlights the number of instances where a particular code is used in the current domain, providing more insight into the use and analysis of the code.

The report output file name is `utsarrp2-<dbname>-<domain>-<date>_<time>.prn`.

Use the Replace Code Utility (`utsarrp3.p`) to correct any issues.

Execute this report separately for each domain that is listed in the Duplicate Code Report with conflicts.

Replace Code Utility (utsarrp3.p)

Use this utility to replace specified code values in the current domain with a new user-specified value.

The report output file name is `utsarrp3-<dbname>-<domain>-<date>_<time>.prn`.

Warning This utility can have a broad effect on the system and future transactions, depending on the codes replaced. Use it with caution and only after backing up the database.

GL Account Type Utility (uxglacup.p)

In QAD Enterprise Edition Financials, each General Ledger (GL) account has an account type, which specifies how the account is used. These types include control accounts, intercompany accounts (known as Cross-Company Control accounts in Enterprise Edition), and bank and cash accounts. The types also include special accounts (classified as System accounts) dedicated to period closing and exchange rate fluctuations. Any other accounts not falling into these specialized classifications are classified as standard accounts. These accounts are predominantly income/expense accounts, but also include some balance sheet accounts as well.

Control, Banking, Cash, Intercompany, and System type accounts have restrictions on how and where they can be used in transactions. Standard accounts have no restrictions other than they cannot be used for purposes associated with the other account types.

System type accounts can have one only one GL account per domain defined for each individual purpose. For example, only one account can be used for unrealized exchange rate gains in a domain, regardless of the currency involved. Similarly, another single GL account is dedicated to exchange rate rounding differences. The one exception to this rule for System type accounts is PO Receipts. Multiple accounts can be used for PO Receipts.

Intercompany accounts are similar to System type accounts, but they are not classified as a System type account. In the old Financials, you can have different intercompany accounts for the debit side and credit side for each functional area (AR, AP, Inventory, and Fixed Assets) and each entity (and in eB2.1 and later for each domain). QAD Enterprise Edition handles this granularity differently.

Separate accounts for intercompany debits and credits are no longer supported. Further, the intercompany account used for a functional area must be the same for that functional area in every entity within the same domain. If this level of detail is not desired, you can use the same Intercompany account for any or all functional areas in a domain.

The utility distinguishes between tax accounts used for AR and AP. However, separate account codes are not required for each of these areas. You can use the same account code for AR and AP taxes if desired.

All banks belonging to the same entity (within the same domain, if applicable) and having the same cash account must also share a payment in process (PIP) account if the Use Payment In Process Acct field is enabled in Accounts Payable Control. Similarly, all banks belonging to the same entity (within the same domain, if applicable) and having the same cash account must share a common drafts payable account if the Use Draft Management field is enabled in Accounts Payable Control. You can use the same account for PIP and drafts payable when there is an overlap of bank code, entity, and cash accounts for PIP and drafts payable. This functionality is possible because both account types are supplier payment accounts in QAD Enterprise Edition. These requirements are only applicable when European Accounting is not being used for the associated domain (if applicable) or database.

For example, consider the following bank definitions, which assume that both fields in Accounts Payable Control are enabled and European Accounting is not used in the domain.

Table 2.4
Bank Definitions

Domain	Entity	Bank	Cash Acct	PIP Acct	Drafts Pay Acct
demo1	1000	AA	1040	2110	2300
demo1	1000	A2	1040	2111	2300
demo1	1000	BB	1041	2110	2300
demo1	1000	B2	1041	2110	2301
demo1	1000	XX	1040	1040	2300
demo1	1000	X2	1040	1041	2301
demo1	2000	CC	1040	1040	2300
demo1	2000	C2	1041	1041	2301

- Banks AA and A2 are in the same domain and entity and use the same Cash and Drafts Payable accounts, but have different PIP accounts. The GL Account Type Utility requires that a single PIP account is designated for these two banks.
- Banks BB and B2 are in the same domain and entity and use the same Cash and PIP accounts, but have different Drafts Payable accounts. The GL Account Type Utility requires that a single Drafts Payable account is designated for these two banks.
- Banks XX and X2 are in the same domain and entity and use the same Cash account, but have different PIP and Drafts Payable accounts. The GL Account Type Utility requires that a single PIP account is designated for these two banks. It also requires that a single Drafts Payable account is designated for these two banks.

  If desired, you can use the same account for any or all of the above PIP and Drafts Payable conflicts.
- No changes are required for the PIP and Drafts Payable accounts in Banks CC and C2. They do not share a Cash account.

GL Allocation codes are no longer permissible in any account type field other than Standard Account types. Even within Standard Accounts, their use is limited. The Data Preparation Report highlights any database fields using a GL allocation code where it is not allowed.

The GL Account Type Utility identifies GL accounts in static data (not transactional data) that do not pass the requirements described previously when moved to QAD Enterprise Edition Financials. For every account type exception encountered in the database, you are prompted for a GL account to use during conversion to correct the exception. Your answer is applied to every database field listed in the detail report for this account type.

When checking for exceptions involving the Accounts Receivable and Accounts Payable control accounts, the utility uses the account information entered through the Control Account Utility as its basis for comparison.

The utility provides a report-only option, listing the GL accounts in conflict. You can run the report in Detail or Summary mode.

- Detail mode lists every instance of an account that is in conflict, the menu function where the conflict occurs, and some key values to aid in identifying the specific offender.
- Summary mode lists only the first instance of an account in conflict and the menu function where it is defined. There can be additional instances other than the one shown.

The report output file name is `uxglacup-dtl-<dbname>-<domain>-<date>_<time>.prn` or `uxglacup-sum-<dbname>-<domain>-<date>_<time>.prn`.

First run the report with Update = No and Detail/Summary = Detail as a planning tool to determine how to resolve the exceptions.

Once this action is done, the corrections can be done in several ways.

- Correct the offending account directly in the program shown on the report.
- Use the GL Account Type Utility to make the correction.
- Use a combination of the above two methods.

QAD recommends using the utility to make corrections. Otherwise, any changes can affect how future transactions are booked to the General Ledger. When using the utility and assigning new accounts to Standard Account types, pay careful attention to the detail report to understand all of the places where the New Acct value is applied.

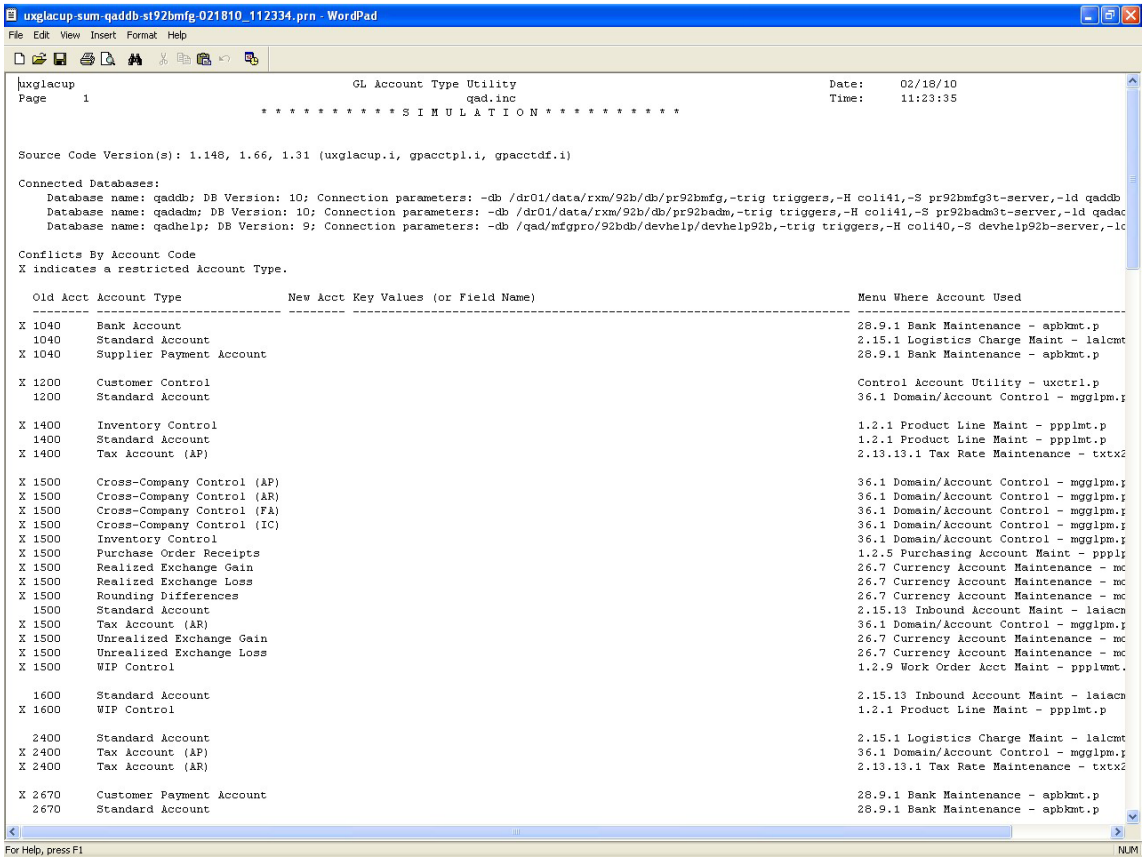
Run the utility as many times as necessary. If it was previously run in Update mode, the previous answers are displayed on the screen. Each time the utility runs, it checks for additional exceptions since the last execution. Therefore, run the utility a final time in Update mode just before starting the conversion.

For eB2.1 and later, run the utility in each active domain. The Data Preparation Report checks and verifies that all exceptions were resolved. If not, errors are reported.

This utility, when run in Update mode, does not affect the system or future transactions. It only creates QAD Work Table (qad_wkfl) records. However, any account conflicts resolved by modifying the accounts directly in their menu functions (for example, Product Line Maintenance, System Control, and so on) affect subsequent transactions involving the account fields updated. Consider this impact when developing a plan to resolve the account conflicts noted on the report this utility produces.

The following report uses account 1040 as an example.

Fig. 2.3
GL Account Type Utility Output



uxglacup-sum-qaddb-st92bmfg-021810_112334.prn - WordPad

File Edit View Insert Format Help

```
uxglacup                                    GL Account Type Utility                                    Date:     02/18/10
Page     1                                          qad.inc                                          Time:     11:23:35
                                  * * * * * * * * * * S I M U L A T I O N * * * * * * * * * *


Source Code Version(s): 1.148, 1.66, 1.31 (uxglacup.i, gpacctpl.i, gpacctdf.i)

Connected Databases:
    Database name: qaddb; DB Version: 10; Connection parameters: -db /dr01/data/rxm/92b/db/pr92bmfg,-trig triggers,-H coli41,-S pr92bmfg3t-server,-ld qaddb
    Database name: qadadm; DB Version: 10; Connection parameters: -db /dr01/data/rxm/92b/db/pr92badm,-trig triggers,-H coli41,-S pr92badm3t-server,-ld qadac
    Database name: qadhelp; DB Version: 9; Connection parameters: -db /qad/mfgpro/92bdb/devhelp/devhelp92b,-trig triggers,-H coli40,-S devhelp92b-server,-lc

Conflicts By Account Code
X indicates a restricted Account Type.

  Old Acct Account Type              New Acct Key Values (or Field Name)                                                                  Menu Where Account Used
  -------- ------------------------- -------- ------------------------------------------------------------------------------------------- ------------------------------------
X 1040     Bank Account                                                                                                                  28.9.1 Bank Maintenance - apbkmt.p
  1040     Standard Account                                                                                                              2.15.1 Logistics Charge Maint - lalcmt
X 1040     Supplier Payment Account                                                                                                      28.9.1 Bank Maintenance - apbkmt.p

X 1200     Customer Control                                                                                                              Control Account Utility - uxctrl.p
  1200     Standard Account                                                                                                              36.1 Domain/Account Control - mgglpm.p

X 1400     Inventory Control                                                                                                             1.2.1 Product Line Maint - ppplmt.p
  1400     Standard Account                                                                                                              1.2.1 Product Line Maint - ppplmt.p
X 1400     Tax Account (AP)                                                                                                              2.13.13.1 Tax Rate Maintenance - txtx2

X 1500     Cross-Company Control (AP)                                                                                                    36.1 Domain/Account Control - mgglpm.p
X 1500     Cross-Company Control (AR)                                                                                                    36.1 Domain/Account Control - mgglpm.p
X 1500     Cross-Company Control (FA)                                                                                                    36.1 Domain/Account Control - mgglpm.p
X 1500     Cross-Company Control (IC)                                                                                                    36.1 Domain/Account Control - mgglpm.p
X 1500     Inventory Control                                                                                                             36.1 Domain/Account Control - mgglpm.p
X 1500     Purchase Order Receipts                                                                                                       1.2.5 Purchasing Account Maint - pppl
X 1500     Realized Exchange Gain                                                                                                        26.7 Currency Account Maintenance - mc
X 1500     Realized Exchange Loss                                                                                                        26.7 Currency Account Maintenance - mc
X 1500     Rounding Differences                                                                                                          26.7 Currency Account Maintenance - mc
  1500     Standard Account                                                                                                              2.15.13 Inbound Account Maint - laiacm
X 1500     Tax Account (AR)                                                                                                              36.1 Domain/Account Control - mgglpm.p
X 1500     Unrealized Exchange Gain                                                                                                      26.7 Currency Account Maintenance - mc
X 1500     Unrealized Exchange Loss                                                                                                      26.7 Currency Account Maintenance - mc
X 1500     WIP Control                                                                                                                   1.2.9 Work Order Acct Maint - ppplwmt.

  1600     Standard Account                                                                                                              2.15.13 Inbound Account Maint - laiacm
X 1600     WIP Control                                                                                                                   1.2.1 Product Line Maint - ppplmt.p

  2400     Standard Account                                                                                                              2.15.1 Logistics Charge Maint - lalcmt
X 2400     Tax Account (AP)                                                                                                              36.1 Domain/Account Control - mgglpm.p
X 2400     Tax Account (AR)                                                                                                              2.13.13.1 Tax Rate Maintenance - txtx2

X 2670     Customer Payment Account                                                                                                      28.9.1 Bank Maintenance - apbkmt.p
  2670     Standard Account                                                                                                              28.9.1 Bank Maintenance - apbkmt.p
```

For Help, press F1 NUM

The instances with an X must have a unique account code separate from the others.

It is not necessary to change each instance of account 1040 to a new account. Nor is it necessary to change every instance of account 1040 with an X to a new account.

The objective is for all three of these instances to have different accounts after the conflicts for 1040 are resolved. Any one of them can retain the original account 1040 as long as neither of the other two is assigned account 1040.

See Table 2.3 on page 19 for the expected category for each Enterprise Edition account type.

Impact of GL Account Type Utility During a Conversion

The data entered in the GL Account Type Utility is used for two purposes during a conversion:

- If there was an account Type conflict before conversion, the data entered in the utility is used to set the Type of the GL Account in the Enterprise Edition Financials.
- A new account is created and assigned to the relevant fields where you have elected to replace an account with a new Account Code for a particular usage.

Pre-conversion Integrity Report (gpinckrp.p)

This report assesses the status of financial transaction data before conversion. It reports AR, AP, and GL transaction integrity information in a single report.

You cannot correct all of the highlighted inconsistencies. Where possible, the necessary tools are noted in the section for each applicable area.

The report is not intended to highlight issues that require correction before conversion. It is used with the Post Conversion Integrity Check (36.16.23.3, `acinckrp.p`) in the Enterprise Edition Financials after conversion to substantiate that the conversion did not alter the integrity of the financial data. See “Post Conversion Integrity Check (36.16.23.3, acinckrp.p)” on page 61.

The report output file name is `gpinckrp-dtl-<dbname>-<date>_<time>.prn` or `gpinckrp-sum-<dbname>-<date>_<time>.prn`.

Setting Build Integrity Check Records to Yes stores data in the database for the current report execution, enabling later reproduction of the report showing the same information. Anytime the status of the database is required, set this field to Yes. Set this field to No to run the report using the data collected from a previous execution.

Report Integrity Check Records displays the information collected during the first step. This field is usually set to Yes.

For eB2.1 or later, the reports span all active domains regardless of the domain where they are launched.

If data is corrected, run this report again to capture an accurate pre-conversion snapshot. Rerun this report as many times as necessary.

AR Transaction Integrity

This portion of the report compares the sum of open AR invoices and payments by account to the sum of the corresponding amounts for each AR GL account. It reports any differences in the local and/or transaction currency amounts. The report also lists any non-AR GL transactions posted against an AR control account. Finally, the sum of open AR invoices and unapplied payments by customer is compared to each customer’s Open Balance in Customer Maintenance, reporting any differences. You can correct differences by running the Adjust Customer Balance Utility (36.25.5, `utcsbal.p`).

It also performs various database integrity checks:

- The customer referenced on each invoice still exists.
- Every invoice has detail line information.

AP Transaction Integrity

This portion of the report compares the sum of open AP vouchers by account to the sum of the corresponding amounts for each AP GL account.

It reports any differences in the local and/or transaction currency amounts. The report also lists any non-AP GL transactions posted against an AP control account. Finally, the sum of open AP vouchers by supplier is compared to each supplier's Open Balance in Supplier Activity Inquiry, reporting any differences. You can correct differences by running the Adjust Supplier Balance Utility (36.25.4, `utvdbal.p`).

It also performs various database integrity checks:

- That the supplier referenced on each voucher still exists.
- That every voucher has detail line information.

GL Transaction Integrity

The totals of posted GL transactions by account in the General Ledger Transaction History (gltr_hist) table are compared to the amounts stored in the Account Balance (acd_det) table. Differences are reported and can be corrected by running the Recalculate acd_det Totals Utility (36.25.39, `utacdfix.p`).

Warning Running the Recalculate acd_det Totals Utility in a database containing consolidated GL transactions zeros-out the period totals for all periods other than the consolidation period.

Any out-of-balance transactions are reported because the conversion must balance them by creating offsetting entries. If the database contains GL transactions consolidated using selected accounts, run the report in Summary mode, which verifies that the transactions are in balance for the year, not by individual transaction.

It also performs these database integrity checks:

- That the entity in each GL transaction still exists.
- That the account for each GL transaction still exists and is active.
- That all effective dates for GL transactions have corresponding periods in the GL calendar.

If any of these types of errors are reported, fix them manually before beginning the conversion.

Set Conversion Parameters

Appendix Conversion Parameters Utility (utfinpar.p)

This utility prompts you for values for the parameters used in the QAD Enterprise Edition conversion when creating new financial objects. Customers converting from eB2.1 and later with multiple domains may want to provide a different set of values for each domain. For eB2.1 and later, run the utility in each active domain, regardless of whether values differ by domain.

Most these parameters are for daybook codes used in various areas of the applications. Other parameters are items such as default credit terms, sub-account, cost center, and so on. You can specify some parameters by domain whereas others are system level and apply to all domains.

Since there are so many parameters, QAD suggests running this utility well in advance of the conversion to allow ample time for planning and decision making. The following table lists the parameters along with a brief explanation of each. eB2.1 and later users could find it helpful for planning purposes to extract this table to a spreadsheet with one column per domain.

If new validations were added to the utility since it was last run, the Data Exists flag for parameters with existing values is set to No. This behavior causes a warning to appear when the utility starts because the existing values are treated as invalid. Attempting to reaccept an invalid parameter displays an error message that explains why the parameter is no longer valid.

Rerun the utility as often as needed and delete or change values.

Table 2.5
New Financial Object Parameters

Level	Parameter	Use
Domain	Default AR Finance Charges Daybook	Default daybook code to use for this daybook type. If Finance Charges are not used, this parameter is not required.
Domain	Default AR Credit Note Daybook for Operational Invoices	Default daybook code to use for this daybook type
Domain	Default AR Credit Note Daybook for Financial Invoices	Default daybook code to use for this daybook type
Domain	Default AR Invoice Daybook for Operational Invoices	Default daybook code to use for this daybook type
Domain	Default AR Invoice Daybook for Financial Invoices	Default daybook code to use for this daybook type
Domain	Default AR Payment Daybook	Default daybook code to use for this daybook type
Domain	Default Customer Adjustment Daybook	Default daybook code to use for this daybook type
Domain	Prefix for AR Tax Register Daybook Sets	The prefix to use when creating AR Tax Register Daybooks Sets
Domain	Prefix for AP Tax Register Daybook Sets	The prefix to use when creating AP Tax Register Daybooks Sets
Domain	Default AP Credit Note Daybook for Financial Invoices	Default daybook code to use for this daybook type

Table 2.5 — *New Financial Object Parameters* (Page 1 of 4)

Level	Parameter	Use
Domain	Default AP Invoice Daybook for Financial Invoices	Default daybook code to use for this daybook type
Domain	Default ERS Supplier Invoice Daybook	Default daybook code to use for this daybook type. This prompt only appears if ERS is used.
Domain	Default ERS Supplier Credit Note Daybook	Default daybook code to use for this daybook type. This prompt only appears if ERS is used.
Domain	Default AP Payment Daybook	Default daybook code to use for this daybook type
Domain	Default Intercompany Daybook	Default daybook code to use for this daybook type. Specify a daybook code for each functional area (IC, FA, Inventory, SO, and WO), but they can the share code if desired.
Domain	Default Matching Entry Daybook	Default daybook code to use for this daybook type
Domain	Default Banking Entry Daybook	Default daybook code to use for this daybook type
Domain	Default Inventory Daybook	Default daybook code to use for this daybook type
Domain	Default Work Order Daybook	Default daybook code to use for this daybook type
Domain	Default Sales Order Daybook	Default daybook code to use for this daybook type
Domain	Default Fixed Assets Daybook	Default daybook code to use for this daybook type
Domain	Default Daybook for Journal Entries	Used for any domain that has not previously used daybooks when converting GL transactions. Note: This parameter is a new daybook code that the conversion creates and not one of the default daybooks specified previously.
Domain	Default Daybook Sequence Effective Date (MM/DD/YYYY)	The earliest date when the new Financial daybooks can be used. This value is also used for operational daybooks if daybooks were not used before the conversion. Therefore, set the date to the earliest of the first open transaction or contract.
Domain	Account for Posting Balances	The conversion uses this account when it balances any unbalanced double-sided GL transactions. The account where such offsets are posted. Note: This is a new account that does not exist in the database. This account is created automatically during the conversion process.
Domain	Account for Year-End Balances	New account the conversion creates for posting the offset to year-end closing entries.

Table 2.5 — *New Financial Object Parameters* (Page 2 of 4)

Level	Parameter	Use
Domain	Account for Results of Current Year	This account is used to accumulate the YTD profit/loss total for printing on the Balance Sheet. You are only prompted for this account if GL transactions have been posted to the Profit/Loss Account (BS) (co_ctrl.co_pl) in General Ledger Control (25.24, `glcopm.p`). If prompted for this account, it must be an account with no transactions posted against it. It can be a new account. If so, the conversion creates it. If you are not prompted for this account, the conversion uses co_ctrl.co_pl.
Domain	Account for Current Year Income Offset	This prompt only appears if the GL Report Writer is not used or no account was assigned for this purpose in that module. If prompted for this account, it must be new and not exist. The conversion creates it.
Domain	Account for Results of Previous Years	The conversion creates this account. It is used to accumulate the profit/loss total for previous unclosed years when printing the Balance Sheet.
Domain	Default Account for Unmatched Invoices	Default GL account to use for Unmatched Invoices. If using European Accounting, the use Waiting Expenses account in European Accounting Control (25.23, `eueupm.p`) defaults to this field and the user cannot modify it. If not using European Accounting, specify a new account to create during conversion.
Domain	Default Sub-Account	Default sub-account to use when an account requires a sub-account. Note: This is a new value that does not exist in the database. See also “COA Mask” on page 156.
Domain	Default Cost Center	Default cost center to use when an account or sub-account requires a cost center. Note: This parameter is a new value that does not exist in the database. See also “COA Mask” on page 156.
Domain	Default Project	Default project code to use when an account, sub-account, or cost center requires a project code. Note: This parameter is a new value that does not exist in the database. See also “COA Mask” on page 156.
Domain	Default Credit Term	Used when an invalid credit term is referenced against a source record.
Domain	Convert Unused Cashbooks?	Should the conversion convert “not used” cashbooks (for example, ba_mstr records where ba_status = “NU”)?

Table 2.5 — *New Financial Object Parameters* (Page 3 of 4)

Level	Parameter	Use
Domain	Create Entity Tax Periods?	Should the conversion create Tax Periods for each entity? If this option is selected, the conversion creates the Tax Periods that are the same as the existing GL Periods.
System	Default Tax Class for Suspended Tax	The default tax class to use for Suspended Tax. This prompt only appears if the European Accounting module is in use. SP3 and above only.
System	Reporting Currency	This parameter is used as the default value for the Reporting Currency in QAD Enterprise Edition Financials.
System	Country for default Tax Box	Country code for the default Tax Box used in Tax Reporting
System	Source directory for Payment Format XML files	The directory from which the Payment format XML files are loaded. This option only appears if the source database uses European Accounting.
System	Convert existing security?	Should the conversion convert the existing security or convert Users, Groups, and allow the user to re-implement the security after the conversion?

Table 2.5 — *New Financial Object Parameters* (Page 4 of 4)

You can exit the utility without providing all of the answers, but you must complete it before running the conversion.

The report output file name is `utfinpar-<dbname>-<domain>-<date>_<time>.prn`.

This utility does not affect the system or future transactions. It only creates QAD Work Table (qad_wkfl) records.

Pre-conversion Completion

Converted GL Account Definitions Report (glacdfrp.p)

This optional report provides a preview of how the Enterprise Edition Financials will define your accounts after conversion. You cannot run this report until you have completed the Control Account Utility, GL Account Type Utility, and GL Project Account/Project Utility (if pre-eB). You also must have entered the account parameters into the Conversion Parameters Utility.

Process Pending Transactions

Process pending transactions as follows:

1. Print any pending invoices (Pending Invoice Print, 7.13.3, `sosorp10.p`).
2. Post any remaining invoices (Invoice Post, 7.13.4, `soivpst.p`).

3 Confirm all unconfirmed vouchers (Voucher Confirmation Automatic, 28.6, `apvoco01.p`; Voucher Confirmation Manual, 28.7, `apvoco.p`).

   Note This step is optional. See "Unconfirmed Supplier Vouchers" on page 181.

4 Post all GL transactions (Transaction Post, 25.13.7, `gltrpst.p`).

Close the Production Database to Users

Use Menu Security (36.10, `mgpwmt.p`) to prevent the addition of new data to the database and further transactions.

Rerun Pre-conversion Integrity Report (gpinckrp.p)

Rerun the Pre-conversion Integrity Report. See "Pre-conversion Integrity Report (gpinckrp.p)" on page 27.

Rerun GL Account Type Utility (uxglacup.p) in Update Mode

1 Rerun the GL Account Type Utility in Update mode (in all active domains for eB2.1 and later). See "GL Account Type Utility (uxglacup.p)" on page 23.

   An error displays if the GTM Conversion was not done and conflicts were found in the accounts from the obsolete tax tables (tax_mstr, vt_mstr). The utility is not flagged as complete until the GTM Conversion is done. Run this utility again to check for conflicts involving accounts in the new tax table Tax Master (tx2_mstr).

2 Resolve any new conflicts.

Run the Converted GL Account Definition Report (glacdfrp.p)

This report provides a preview of how the GL accounts and their key attribute settings will look after the conversion. Use the output from this report to more accurately plan and prepare a company's GL account data before conversion execution.

You can only execute this report after you run the following utilities:

- Control Account (`uxctrl.p`)
- GL Account Type (`uxglacup.p`) (in update mode)
- Conversion Parameters (`utfinpar.p`)
- GL Account Project Code (if pre-eB)

The report is provided for informational purposes and is not mandatory. For versions eB2.1 and later, the report content is specific to the domain in which the report was initiated.

This report has no effect on the system or future transactions. You can run it as often as desired.

Standard Period Closing Reports

Although their use is optional and does not directly affect the conversion, QAD recommends that you run the standard period closing reports for later comparison with the equivalent post-conversion reports. This action helps ensure that the conversion did not change the data. The standard period closing reports are as follows:

- Trial Balance (25.15.4, `gltbrp.p` or 25.15.5, `gldtbrp.p`)
- Balance Sheet (25.15.8, `glbsrp.p`)
- Income Statement (25.15.13, `glinrp.p`)
- AR Aging as of Effective Date (27.18, `arcsrp05.p`)
- AP Aging as of Effective Date (28.17.6, `apvorp04.p`)
- AR Tax by Tax Rate (2.13.15.14, `txarrp01.p`)
- AP Tax by Tax Rate (2.13.15.17, `txaprp01.p`)
- AR Balance Report (27.20, `arcsrp.p`)
- Inventory Valuation (3.6.13, `ppptrp03.p`)
- Unvouchered Receipts Report (5.13.10, `poporp11.p`)
- Purchase Receipt Report (5.13.5, `poporp06.p`)
- Asset Owned Report (32.5.11, `faaorp.p`)
- Sales Order Report (7.15.1, 2 or 3, `sosorp.p`, `sosorp01.p`, `sosorp02.p`)

Rerun (Finalize) Data Preparation Report

Rerun the Data Preparation Report to ensure that no errors were introduced. Correct the data until no errors are present.

Chapter 3

Conversion Execution

This chapter describes how to execute a Progress database conversion.

Introduction

This section provides detailed instructions on how to execute the conversion.

Important Only use the following conversion instructions to convert a Pre-Enterprise Edition environment to Enterprise Edition. Do not use this option if your current environment is Enterprise Edition. For EE upgrades, see Chapter 5, “Upgrading QAD Enterprise Edition into a New Environment,” on page 73 or Chapter 6, “Upgrading an Existing QAD Enterprise Edition Environment,” on page 81.

Note QAD recommends that you run pre-EE to EE conversions without Financials patches installed. Pre-EE to EE conversions are certified against the initial 2022EE release. The conversion process includes Financials conversions, which are tightly linked to the Financials software. Doing a conversion with Financials patches installed can lead to additional issues.

Prerequisites

The Pre-EE to EE conversion software is not deployed with the standard Enterprise Edition media but as an independent image (`yab-conv-pre-ee`) that you must download. For more information, please contact QAD Support or your Account Manager.

Converting to Enterprise Edition

The conversion process consists of the following major tasks:

- Back Up the Environment Databases
- Preserve Custom System Data
- Prepare Source Databases
- Install Enterprise Edition
- Conversion Prerequisites
- Configure the Conversion
- Execute the Conversion
- Enter License Codes
- Conduct Conversion Execution Troubleshooting
- Rerun the Conversion Execution

Back Up the Environment Databases

Before starting conversion activities, always back up the existing environment databases. Refer to the Progress documentation at www.progress.com/support for Progress backup procedures.

Preserve Custom System Data

The conversion does not convert custom data changes. If you have added custom data to any of the default system data listed below, you must dump the customized records before beginning a conversion.

Note When dumping data from a UTF-8 database, Progress does not support character UTF-8 clients. Use a binary dump or create the relevant query that can be executed in batch mode.

- Custom menus and messages
- Custom labels
- Custom language detail
- Custom field help lookup
- Custom reports, browses, and favorites used in the QAD .NET UI

If you have not made any customizations, skip this section.

After dumping your custom data and completing your conversion, incorporate your customizations in your target version environment.

Prepare Source Databases

If the Progress version of the databases (qaddb and qadadm) differs from the Progress version in the new QAD product environment, convert the databases to the new Progress version using the Progress `proutil` utility.

For additional information about the `proutil` utility, see the *Progress OpenEdge Data Management: Database Administration* guide.

Conversion execution will not start if there are errors in the Data Preparation Report. Resolve all pre-conversion errors before proceeding with the conversion.

Note The user `mfg` must exist within the source database and have a blank password. This is required for the conversion to successfully start a Financials session when executing the SE to EE Financial conversions.

Install Enterprise Edition

The first step in the conversion process is to install the latest version of Enterprise Edition. For detailed information about QAD Enterprise Edition installation, see QAD Enterprise Edition Installation Guide.

Note If the source database uses large files, the target database should also be configured to use large files.

Important When installing QAD Enterprise Edition, do not install demo data. If you have already installed demo data, you must remove it. See the "Using Demo Data" section of the *QAD Administration and Configuration Guide* for YAB 1.15 on the QAD Document Library.

Conversion Prerequisites

The following are required before executing a Pre-EE to EE Conversion.

Table 3.1
Conversion Prerequisites

Name	Package Name	Minimum Version	Description
YAB Pre EE Conversions Enabled	yab-conv-pre-ee	1.15.0.x	A package that contains the process to execute a pre-EE to EE conversion. ***Note***: This package is restricted to approved customers only.

After downloading and saving the packages to a suitable location, install them with the following commands:

```
> yab -install-update:false install /<package path>/yab-conv-pre-ee-
1.15.x.x.zip
```

Configure the Conversion

The conversion-pre-ee-upgrade process requires you set one mandatory configuration property. This setting, `conv.pre-ee.src-db.dir`, must be defined in the `configuration.properties` file before executing the conversion. For example:

```
conv.pre-ee.src-db.dir=/dr01/srcdb/
```

Note To avoid any lock table overflow errors, especially for large databases, you may need to increase the Lock Tables Entries value for the QADDB. It is recommended that you increase the `locktableentries` and apply the change as follows:

a Update `configuration.properties`:

```
dbserver.qaddb.locktableentries=327680
```

b Apply the change:

```
> yab reconfigure
```

Table 3.2
Conversion Configuration

Properties	Default	Description
conv.pre-ee.src-db.dir		The directory where source QAD and Admin databases are located.
conv.pre-ee.src-db-qaddb.physicalname=mfgprod	mfgprod	The unqualified physical name of the source QADDB database.
conv.pre-ee.src-db-qadadm.physicalname=	admprod	The unqualified physical name of the source Admin database.

Note The default physical names of the source databases are **mfgprod** and **admprod**. Use these default names when restoring the source databases to avoid having to override them in the `configuration.properties` file.

Optional Configuration

The conversion-pre-ee-upgrade process has a number of optional configuration properties that are detailed in Table 3.3. These properties have default values and do not need to be explicitly set.

Table 3.3
Optional Configuration Values

Properties	Default	Description
conv-ui.ACDATE	None	ECommerce Turnaround data last accessed date. If blank the conversion uses the current date; that is, today.
conv-ui.ACUSER	mfg	ECommerce Turnaround data last accessed User ID. A value is required.
conv-ui.CRDATE	None	ECommerce Turnaround data creation date. If blank the conversion uses the current date; that is, today.
conv-ui.CRUSER	mfg	ECommerce Turnaround data creation UserID. A value is required.

Properties	Default	Description
conv-ui.costpoint01	Usage	Designate the source of the item PO cost to use for supplier consigned purchase orders in Accounts Payable. Only a value of Usage or Receipt is permitted.
conv-ui.dom_domain	test	Only used when converting from source version 92a or earlier (86e, 90, 91, 92, 92a). No validation.
conv-ui.dom_name	conversion	Required when converting from 86e, 90, 91, 92, 92a. No validation.
conv-ui.dom_sname	test	Required when converting from 86e, 90, 91, 92, 92a. No validation.
conv-ui.dbtype	Progress	The database type, which by default is Progress. Only a value of Progress is permitted.
conv-ui.prlistassignto	1	Designate existing price lists used for scheduled orders and/or RMA receipts as Customer or Supplier price lists. Only a value of 1 or 2 is permitted.

For example, to change `conv-ui.costpoint01` to “Receipt” from its default value of “Usage” add the following line to `configuration.properties`:

```
conv-ui.costpoint01=Receipt
```

To view more detailed information for each configuration setting, use the yab help command.

For more information regarding making configuration changes, refer to the QAD Adaptive ERP Configuration and Administration Guide.

Execute the Conversion

Executing a conversion requires four steps.

1 Prepare Adaptive UX Environments

2 Execute the Conversion Process

3 Validate the Conversion Execution

4 Load System Data

Important The `conversion-pre-ee-upgrade` process executes the `database-qaddb-rebuild` and `database-qadadm-rebuild` processes, which delete and re-create the qaddb and qadadm databases. Any data in your EE databases will be deleted during this process and all manual changes you made to the databases will need to be reapplied. For example, if you have Transparent Data Encryption enabled for database security, you must re-enable TDE after the conversion process is complete.

Prepare Adaptive UX Environments

If you are executing the conversion on an Adaptive ERP environment that includes Adaptive UX, you must perform an additional step before executing the conversion process. If your environment does not include Adaptive UX, skip to the next step, Execute the Conversion Process.

Reset the Adaptive UX conversion by deleting the following state file. Enter:

```
> rm <appdir>/build/work/system/process-ignore
```

Execute the Conversion Process

The conversion is started with the following yab command:

```
> yab conversion-pre-ee-upgrade
```

The conversion proceeds until it completes or an error occurs. No further input is required until the conversion ends.

The conversion can be monitored using the following log files:

```
> ../<application-directory>/logs/yab.log
> ../<application directory>/logs/conversion.log
```

Once the conversion is complete, review the log files to ensure there were no errors reported. See Appendix E, “Log Files,” on page 189 for information about log files.

Note Conversions are complex processes that can terminate prematurely when data-related errors are encountered. If the conversion terminates prematurely, determine the root cause of the failure and correct it in the source environment. Then restart the conversion from the beginning or the conversion snapshot.

Note QAD recommends that you take a conversion snapshot during the conversion execution. See Appendix G, “Snapshots,” on page 197 for more details on executing a conversion and taking conversion snapshots.

Note See Appendix F, "AIM Conversion," on page 193 for details on executing the AIM database conversion.

Validate the Conversion Execution

The conversion progress can be monitored by examining the following log files:

```
> ../<application-directory>/logs/yab.log
> ../<application directory>/logs/conversion.log
```

Once the conversion has completed, the following checks should be done.

Review Console Output – Build Successful

Ensure the `conversion-pre-ee-upgrade` process completed successfully. The output should be similar to the following.

Note The timings vary, depending on the size of the database and the server specification.

```
host: /dr02/qadapps/ee $ yab conversion-pre-ee-upgrade
                conversion-pre-ee-upgrade (85 tasks)
-----------------------------------------------------------------------
1/85  conversion-preconv-check                        OK (0.823 s)
2/85  conversion-stage                                OK (0.159 s)
3/85  conversion-mfg-package-stage                    OK (6.973 s)
4/85  conversion-fin-package-stage                    OK (6.182 s)
5/85  conversion-mfg-get-src-db-version               OK (0.281 s)
6/85  conversion-fin-get-src-db-version               OK (0.010 s)
7/85  conversion-mfg-create-execution-list            OK (0.665 s)
8/85  conversion-fin-create-execution-list            OK (0.417 s)
9/85  conversion-fin-src-db-data-dump                 SKIPPED (0.010 s)
10/85 conversion-mfg-src-db-data-dump                 OK (1.237 s)
11/85 conversion-domain-data-create                   OK (0.007 s)
...
56/85 conversion-qadadm-index-deactivate              OK (1.131 s)
57/85 conversion-src-admdb-buffercopy                 OK (3.918 s)
58/85 database-qadadm-index-rebuild                   OK (8.161 s)
59/85 conversion-qaddb-index-deactivate               OK (2:05 m)
60/85 conversion-src-db-buffercopy                    OK (2:04 m)
61/85 database-qaddb-index-rebuild                    OK (1:22 m)
62/85 conversion-tlc-create                           OK (4.083 s)
...
80/85 conversion-limited-sync-run                     OK (20.199 s)
81/85 conversion-mfg-pre-ee-qaddb-run                 OK (46.628 s)
82/85 conversion-fin-pre-ee-qaddb-run                 OK (11:23 m)
83/85 conversion-qadaim-run                           OK (0.000 s)
84/85 conversion-mfg-qadadm-run                       OK (1.742 s)
85/85 fin-sync-run                                    OK (2:17 m)
-----------------------------------------------------------------------

BUILD SUCCESSFUL (31:03 m)
```

Note The processes listed above are conversion-specific processes. Examining the output for each of these processes is a good starting point in confirming or troubleshooting the `conversion-pre-ee-upgrade` process.

Review yab.log Output

You should thoroughly review each process associated with the `conversion-pre-ee-upgrade` process in the `yab.log` file. To help navigate the `yab.log` file, you can use the following text to find each process execution entry “BuildContext - ”.

To locate the log content associated with determining the source database version, search the `yab.log` file for:

```
"BuildContext - conversion-mfg-get-src-db-version"
```

To locate the log content associated with buffer copy step, search the `yab.log` file for:

```
"BuildContext - conversion-src-db-buffercopy"
```

To locate the log content associated with qaddb conversion steps, search the `yab.log` file for:

```
"BuildContext - conversion-mfg-pre-ee-qaddb-run"
"BuildContext - conversion-fin-pre-ee-qaddb-run"
```

Review conversion.log Output

You should thoroughly review each conversion process in the corresponding `conversion.log` file. To help navigate the `conversion.log` file, you can use the text in Table 3.4 to find each process execution entry that involves “Executing step:”.

For example, to locate the log content associated with determining the source database version, search the `conversion.log` file for:

```
"Executing step: get-db-version.p"
```

Table 3.4 lists the relationship between the YAB process and corresponding log entry in the `conversion.log` file.

Table 3.4
Conversion.log File Entries

YAB Process	conversion.log Entry
conversion-mfg-create-execution-list	Executing step: create-sel-conv.p
conversion-fin-create-execution-list	Executing step: create-sel-conv.p
conversion-mfg-src-db-data-dump	Executing step: special-dump.p
conversion-fin-src-db-data-dump	Executing step: special-dump.p
conversion-src-admd-buffercopy	Executing step: run-buffer-copy-src.p
conversion-src-db-buffercopy	Executing step: run-buffer-copy-src.p
conversion-mfg-pre-ee-qaddb-run	Executing step: run-qaddb-conv.p
conversion-fin-pre-ee-qaddb-run	Executing step: run-qaddb-conv.p
conversion-mfg-qadadm-run	Executing step: run-qadadm-conv.p

Note If you encounter errors during the conversion, first analyze and resolve the early errors and then repeat the conversion execution. Often, the early errors lead to data not being created, which then causes subsequent errors. By resolving the earlier errors, you often can resolve those related errors.

Load System Data

Once the upgrade has successfully completed, reload the system data with the following command:

```
> yab -clean update
```

Enter License Codes

1. Navigate to the scripts directory and run the client script for the desired language:
   ```
   cd <application_directory>/scripts/client-us.sh
   ```
2. Log in to the system using the following entries:
   - User: mfg
   - Password: [blank]
   - Domain: QAD
3. When prompted, select Add and provide the applicable license codes.

Conduct Conversion Execution Troubleshooting

For information on troubleshooting conversion problems, see Appendix D, “Conversion Troubleshooting,” on page 183.

Rerun the Conversion Execution

The `conversion-pre-ee-upgrade` process records state information to ensure that already-executed processes are not re-executed. If you want to rerun the conversion from the beginning, run the `conversion-pre-ee-upgrade-reset` process and then the `conversion-pre-ee-upgrade` process.

```
> yab conversion-pre-ee-upgrade-reset
> yab conversion-pre-ee-upgrade
```

Note As of YAB 1.14, the `conversion-pre-ee-upgrade-reset` process is executed automatically during the `yab update` step.

Chapter 4

Post-conversion

This chapter describes the validation and set up activities following database conversion.

Introduction

This chapter describes the validation and setup activities following a database conversion from a pre-Enterprise Edition version to Enterprise Edition (these activities are not required when upgrading an existing Enterprise Edition installation).

Post-conversion has the following activities:

- Test and validate a converted database.
- Prepare a converted database for a go-live implementation.

Post-conversion activities consist of the following major tasks:

- Post-conversion Utilities
- Post-conversion Data Validations
- Post-conversion Reports
- Process Flow and Static Data Validation
- Static Data Validation
- Mandatory Post-conversion Set Up
- Optional Post-conversion Set Up

Post-conversion Utilities

Fixed Assets Migration Utility (32.25.2, facvmt.p)

Only run this utility if you are converting from MFG/PRO 8.6E or 9.0 and using the Fixed Assets module. See Appendix B, “Running the Fixed Assets Migration Utility,” on page 137 for more information.

Table Extension Domain Conversions – Part 2

This utility completes the conversion of supplier lot data previously held in table extension records. Run the appropriate menu option for each active domain in the system.

If you are converting to QAD Enterprise Edition from eB2 or eB2.1 with any release before SP 6, run this utility from menu 5.25.7 (`utvdtr92.p`).

If you are converting to QAD Enterprise Edition from eB2.1 SP 6 or higher, run this utility from menu 5.25.8 (`utvdt92b.p`).

Warning Execute one menu item only.

Sales Order Balance Update (36.16.23.6, utcsob.p)

This utility recalculates the open sales order balance by customer and updates the outstanding balance on the customer master. The balance is used when a credit check is performed and the Include Sales Order box is selected.

Document Credit Terms Update (36.25.83, uxdoccrterms.p)

This utility updates various documents (calls, invoice history, sales orders, purchase orders, service contracts, price list detail, sales quotes) of your choosing with a user-specified credit terms code. This credit terms code is applied to documents meeting the selection criteria that have blank or invalid credit terms assigned to them. Blank or invalid credit terms are not converted.

Run the Document Credit Terms Update utility for each domain and separately for each document type.

Update Contract Revenue Account (11.5.13.25.5, utsarrv.p)

This utility moves the Deferred and Accrued Revenue accounts for Service Contracts from the QAD fields to the standard schema fields. If values do not exist in the QAD fields (pre-eB2.1 SP5), this utility locates the appropriate values to use. Run this utility for all versions after you have converted to EE.

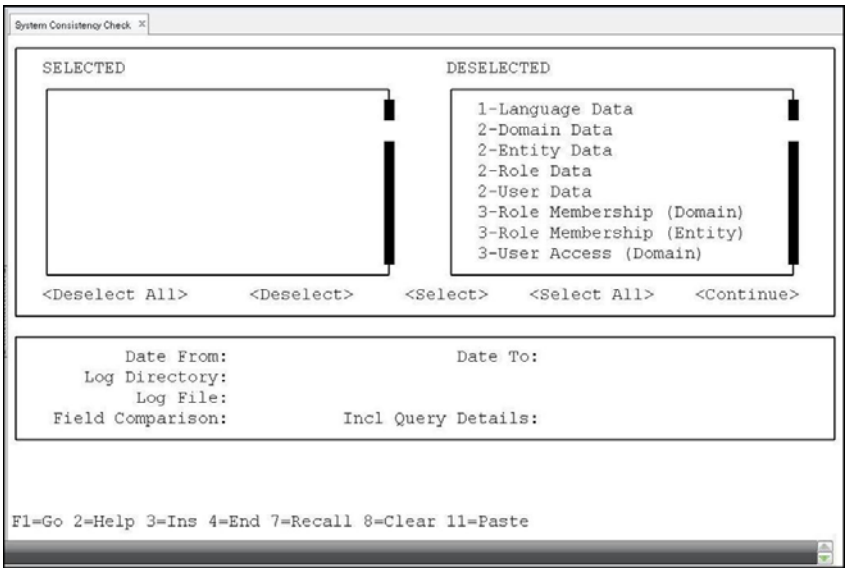
Data Validation

System Consistency Check (36.16.23.1, utsyscon.p)

This report ensures that master data, which is maintained in the Financials and replicated to the operational modules, remains consistent. For example, if there is a record in the operational part of the system, there must be a record in the Financials and the other way around. This check ensures that the conversion created new records in the Financials for every master record in the original version.

1 Start the System Consistency Check report.

Fig. 4.1
System Consistency Check



The resulting report lists any inconsistencies.

Fig. 4.2
System Consistency Check Report

```
2009-05-12>02:49:21: REPORT STATISTICS
                              White Missing * MFGPRO Missing * Inconsistent
        ok 1-Language Data                0 *              0 *            0
        ok 2-Domain Data                  0 *              0 *            0
        ok 2-Entity Data                  0 *              0 *            0
        ok 2-Role Data                    0 *              0 *            0
        ok 2-User Data                    0 *              0 *            0
        ok 3-Role Membership (D           0 *              0 *            0
        ok 3-Role Membership (E           0 *              0 *            0
        ok 3-User Access (Domai           0 *              0 *            0
        ok 3-User Access (Entit           0 *              0 *            0
       ERR 4-Menu Data                    0 *              1 *            0
       ERR 5-Address Data                 0 *              8 *            0
        ok 5-Country Data                 0 *              0 *            0
        ok 5-Credit Terms Data            0 *              0 *            0
        ok 5-Credit Terms Stage           0 *              0 *            0
        ok 5-Currency Data                0 *              0 *            0
       ERR 5-Exchange Rate Data           9 *              9 *            0
        ok 5-GL Calendar Data             0 *              0 *            0
        ok 5-GL Period Data               0 *              0 *            0
        ok 5-Rounding Method Da           0 *              0 *            0
        ok 6-Cost Center Data             0 *              0 *            0
        ok 6-Daybook Data                 0 *              0 *            0
        ok 6-GL Account Data              0 *              0 *            0
        ok 6-GL Mask Data                 0 *              0 *            0
        ok 6-Project Data                 0 *              0 *            0
        ok 6-Sub-Account Data             0 *              0 *            0
        ok 6-Tax Data                     0 *              0 *            0
        ok 7-Customer Data                0 *              0 *            0
       ERR 7-Employee Data               80 *             12 *            0
       ERR 7-End User Contact D           9 *              0 *            2
       ERR 7-End User Data                1 *              0 *            0
        ok 7-Supplier Data                0 *              0 *            0
       ERR 8-AP Matching Data            82 *              0 *            0
       ERR 8-AR Invoice Data              0 *             30 *            0
       ERR 8-Posting Data                 0 *             83 *            0

2009-05-12>02:49:21: >>>> Finished System Consistency Check <<<<
```

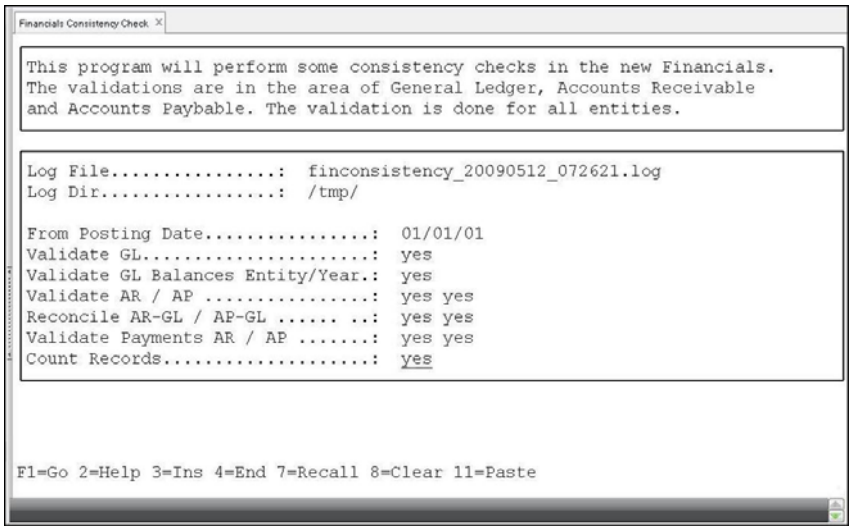
2 Analyze any inconsistencies.

Financials Consistency Check (36.16.23.2, utfincon.p)

This report validates the consistency, completeness, and integrity of converted financial data. It also confirms that all table structures in the new Financials are in order and that their links are correct. This report validates some balances (AR, AP) to ensure transactions balance across entities with regards to debits and credits and that the sub-ledgers balance to GL.

1 Start the Financial Consistency Check report.

Fig. 4.3
Financials Consistency Check



The resulting report contains the details of any inconsistencies.

Fig. 4.4
Financials Consistency Check Report

```
2009-05-11>04:03:34: >>>> START COUNTING <<<<
        Posting#= 1410
        PostingLine#= 3099
        PostingHist#= 477
        QPostingLine#= 0

        DInvoice#= 42
        DInvoiceMovement#= 48
        QDInvoiceMovement#= 0
        DHist#= 28

        CInvoice#= 48
        CInvoiceMovement#= 85
        QCInvoiceMovement#= 0
        CHist#= 29

        CDocument#= 36
        CCollection#= 8
        CDocumentInvoiceXref#= 37
        CDocInvoiceXrefStage#= 0

        DDocument#= 10
        DCollection#= 6
        DDocumentInvoiceXref#= 10
        DDocInvoiceXrefStage#= 0
2009-05-11>04:03:34: >>>> END COUNTING <<<<

2009-05-11>04:03:34: >>>> Start Validation of GL (01/01/00) <<<<<
        POSTING;590842;Posting not in balance for CC amounts;04/26/09;mfg
        PostingLine;591858;Transactions without sub-accounts found on account which is enabled for sub-accounts; 04/26/09;mfg
        PostingLine;591872;Transactions without sub-accounts found on account which is enabled for sub-accounts; 04/26/09;mfg
        PostingLine;590887;Transactions on costcentres found on account which is disabled for costcentres; 04/26/09;mfg
        PostingLine;590987;Transactions on costcentres found on account which is disabled for costcentres; 04/26/09;mfg
        PostingLine;591885;Transactions without sub-accounts found on account which is enabled for sub-accounts; 04/26/09;mfg
        PostingLine;591958;Transactions without sub-accounts found on account which is enabled for sub-accounts; 04/26/09;mfg
        PostingLine;591946;Transactions without sub-accounts found on account which is enabled for sub-accounts; 04/26/09;mfg
        PostingLine;590900;Transactions on costcentres found on account which is disabled for costcentres; 04/26/09;mfg
        PostingLine;591227;Transactions on costcentres found on account which is disabled for costcentres; 04/26/09;mfg
        PostingLine;591203;Transactions on costcentres found on account which is disabled for costcentres; 04/26/09;mfg
```

2 Analyze any inconsistencies.

Operational Account Structure Validation (36.9.20, uxacval.p)

Use this report to generate a report of all of the places in the system where default GL accounts, sub-accounts, and cost centers can be specified and are used to generate GL transactions in the operational modules (for example, Product Line Maintenance, Department Maintenance). The report indicates any invalid combinations.

Invalid combinations can exist for a number of reasons, but more typically occur when accounts, sub-accounts, and cost centers are made inactive. Such invalid settings can result in GL transactions that cannot be posted to the Financials.

1 Run the Operational Account Structure Validation report for each active domain.

Fig. 4.5
Operational Account Structure Validation Menu

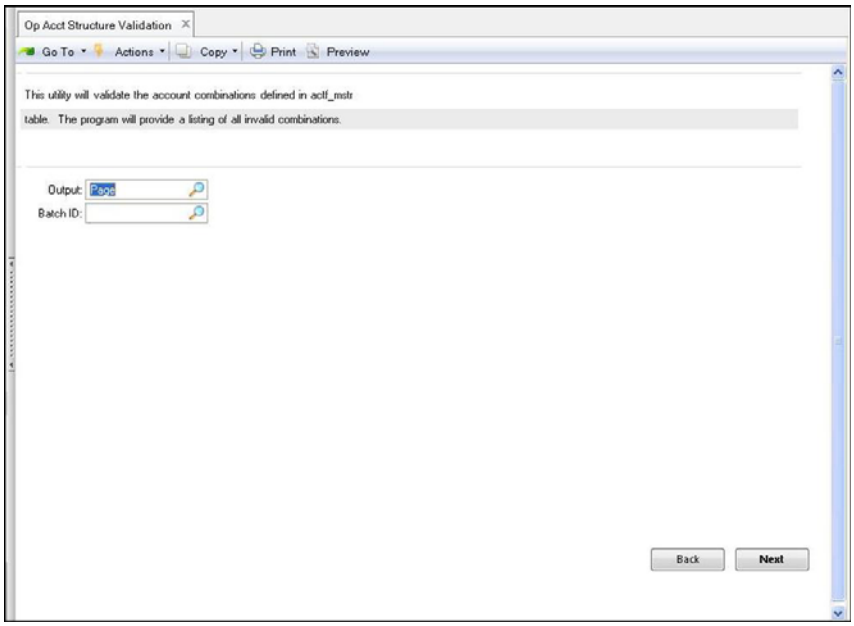


Fig. 4.6
Operational Account Structure Validation Report

2 Analyze the output from this report.

Correct invalid combinations to prevent the generation of invalid GL transactions.

Post Conversion Integrity Check (36.16.23.3, acinckrp.p)

Use this report to produce a post-conversion report to assess the status of the financial transaction data after conversion.

Note Run the corresponding pre-conversion report (`gpinckrp.p`) before conversion. Otherwise, the content of the report is inaccurate.

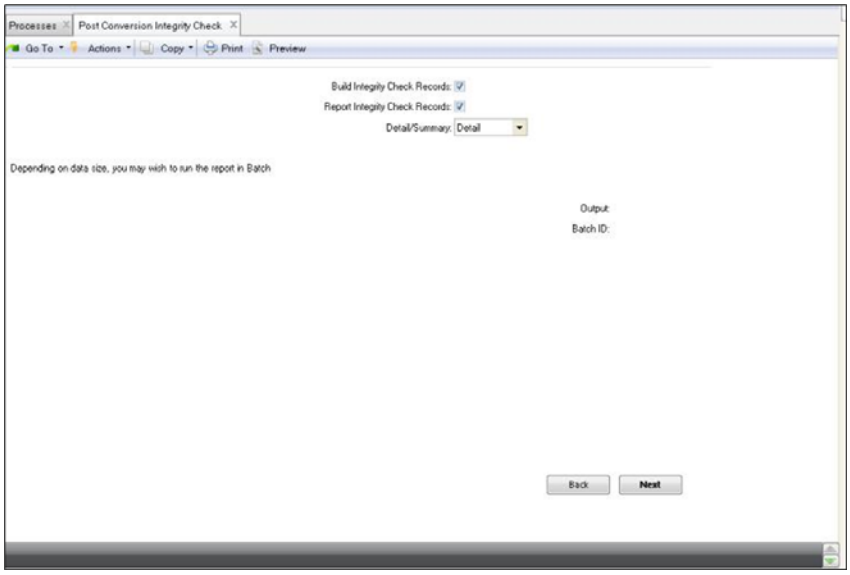
The report is split across domains with one section per domain. Each of the domain sections is broken down into the following subsections:

- GL Transaction Integrity
- AR Transaction Integrity
- AP Transaction Integrity

GL Integrity has two modes. Summary mode only displays accounts containing differences. Detail mode shows all accounts, even when there are no differences between pre- and post-conversion balances.

1 Run the Post Conversion Integrity Check report.

Fig. 4.7
Post-conversion Integrity Check Menu



The report should show that the data is in the same condition as before conversion. Unbalanced GL transactions from the current period, which the conversion balances, will be different.

Fig. 4.8
Post-conversion GL Integrity (Summary)

```
Domain:  Domain1

GL Balance Validation
   Entity    Account     Balance Preconv   Balance Postconv        Difference Conversion Account
   --------- -------- ------------------ ------------------ ------------------ ------------------
   6000      440000              407.72             407.74               0.02 QAD-0086
   6000      410000              100.00             100.05               0.05 QAD-0085
   6000      430000              404.87             404.84               0.03 QAD-0086
   6000      460000              301.99             309.00               8.99 QAD-0086
   6000      480000             1024.00            1022.55               2.45 QAD-0086
```

Fig. 4.9
Post-conversion GL Integrity (Detail)

Domain: Domain1

GL Balance Validation

Entity	Account	Balance Preconv	Balance Postconv	Difference	Conversion Account
6000	000000	0.00	0.00	0.00	
6000	003500	0.00	0.00	0.00	
6000	022000	0.00	0.00	0.00	
6000	023000	0.00	0.00	0.00	
6000	050100	108,267.00	108,267.00	0.00	
6000	051100	721,164.70	721,164.70	0.00	
6000	053100	3,865,197.00	3,865,197.00	0.00	
6000	053900	0.00	0.00	0.00	
6000	054000	92,021.00	92,021.00	0.00	
6000	055000	123,326.00	123,326.00	0.00	
6000	056000	0.00	0.00	0.00	
6000	056100	114,251.00	114,251.00	0.00	
6000	056200	431,145.00	431,145.00	0.00	
6000	059000	0.00	0.00	0.00	
6000	070000	4,383,915.00	4,383,915.00	0.00	
6000	073000	0.00	0.00	0.00	
6000	075100	207,514.00	207,514.00	0.00	
6000	075200	0.00	0.00	0.00	

Fig. 4.10
Post-conversion AP Integrity

AR Account Validation

Entity	Account	Balance Preconv	Conv Acct Balance	Conversion Account	Balance Postconv	Difference
6000	240000	0.00	0.00	QAD-0062	0.00	0.00
6000	240001	0.00	0.00	QAD-0063	0.00	0.00
6000	240002	0.00	0.00	QAD-0064	0.00	0.00
6000	250003	0.00	0.00	QAD-0065	0.00	0.00
6000	250005	0.00	0.00	QAD-0066	0.00	0.00
6000	250007	0.00	0.00	QAD-0067	0.00	0.00
6000	250010	21,669.62	0.00	QAD-0068	21,669.62	0.00
6000	250012	0.00	0.00	QAD-0069	0.00	0.00
6000	250014	0.00	0.00	QAD-0070	0.00	0.00
6000	250017	0.00	0.00	QAD-0071	0.00	0.00
6000	250030	0.00	0.00	QAD-0072	0.00	0.00
6000	250032	0.00	0.00	QAD-0073	0.00	0.00
6000	250041	0.00	0.00	QAD-0074	0.00	0.00
6000	250042	0.00	0.00	QAD-0075	0.00	0.00
6000	250043	0.00	0.00	QAD-0076	0.00	0.00
6000	250044	12,936.00	0.00	QAD-0077	12,936.00	0.00
6000	250045	0.00	0.00	QAD-0078	0.00	0.00
Total		34,605.62	0.00		34,605.62	0.00

Fig. 4.11
Post-conversion AR Integrity

AP Account Validation

Entity	Account	Balance Preconv	Conv Acct Balance	Conversion Account	Balance Postconv	Difference
6000	440000	-36,407.68	0.00	QAD-0025	-36,407.68	0.00
6000	440001	0.00	0.00	QAD-0026	0.00	0.00
6000	440002	-2,315,203.57	0.00	QAD-0027	-2,315,203.57	0.00
6000	465012	-830.00	0.00	QAD-0034	-830.00	0.00
6000	465013	0.00	0.00	QAD-0035	0.00	0.00
6000	465014	-231,902.19	0.00	QAD-0036	-231,902.19	0.00
6000	465015	0.00	0.00	QAD-0037	0.00	0.00
6000	465016	0.00	0.00	QAD-0038	0.00	0.00
6000	465017	0.00	0.00	QAD-0039	0.00	0.00
6000	465018	-30,844.61	0.00	QAD-0040	-30,844.61	0.00
6000	465020	0.00	0.00	QAD-0041	0.00	0.00
6000	465030	0.00	0.00	QAD-0042	0.00	0.00
6000	465031	0.00	0.00	QAD-0043	0.00	0.00
6000	465032	-604.48	0.00	QAD-0044	-604.48	0.00
6000	465039	0.00	0.00	QAD-0051	0.00	0.00
6000	465040	-72,888.38	0.00	QAD-0052	-72,888.38	0.00
6000	465046	0.00	0.00	QAD-0058	0.00	0.00
6000	465047	0.00	0.00	QAD-0059	0.00	0.00
6000	465048	0.00	0.00	QAD-0060	0.00	0.00
6000	465049	0.00	0.00	QAD-0061	0.00	0.00
Total		-2,688,680.91	0.00		-2,688,680.91	0.00

2 Analyze the report output.

Post-conversion Reconciliation Reports

AP Reconciliation Report (36.16.23.10, appcrnrp.p)

Use this optional report to the report pre- and post-conversion balances of open AP transactions.

The report displays base and transaction currency balances and identifies any discrepancies between the pre- and post-conversion values. The output options are Detail, which shows all open transactions, or Summary, which shows only unbalanced transactions. The report can be sorted by Supplier, Effective Date, or Transaction Type, and can be executed for a specific domain or all domains.

Note You initially run the report before beginning transaction processing within the AP module of the converted database. Otherwise, the content of the report output is inaccurate.

During the initial run of the report, each transaction is flagged as reconciled or unreconciled. In subsequent runs of the report, the balances for unreconciled transactions are updated to account for corrections made. The transactions are flagged as reconciled when appropriate.

AR Reconciliation Report (36.16.23.9, arpcrnrp.p)

Use this optional report to report the pre- and post-conversion balances of open AR transactions.

The report displays base and transaction currency balances and identifies any discrepancies between the pre- and post-conversion values. The output options are Detail, which shows all open transactions, or Summary, which shows only unbalanced transactions. The report can be sorted by Customer, Effective Date, or Transaction Type and can be executed for a specific domain or all domains.

Note You initially run the report before starting transaction processing within the AR module of the converted database. Otherwise, the content of the report output is inaccurate.

When the report is first run, each transaction is flagged as reconciled or unreconciled. During subsequent runs of the report, the balances for unreconciled transactions are updated to account for corrections made. The transactions are flagged as reconciled when appropriate.

Post-conversion Reports

Rerun the following reports and compare the pre- and post-conversion data:

- Trial Balance (25.15.4, `gltbrp.p` or 25.15.5, `gldtbrp.p`)
- Balance Sheet (25.15.5.4)
- Income Statement (25.15.5.5)
- Customer Aging Analysis
- Supplier Aging Analysis
- Inventory Valuation as of Date
- Unmatched PO Receipts as of Date (5.13.10)
- Asset Owned Report (Fixed Assets Valuation) (32.5.11, `faaorp.p`)
- Open Sales Order Balances

For more information on post-conversion reports, see *QAD Financials User Guide*.

Process Flow and Static Data Validation

Following conversion, QAD recommends that you process the entire inbound, outbound, and manufacturing transaction life cycles. Review the results at each step to ensure that they are as expected.

Static Data Validation

Following conversion, take a sampling of static data (for example, customer and supplier addresses) and verify that everything is as expected.

Mandatory Post-conversion Set Up

Warning For eB2.1 and later, review county and state data before conversion to eliminate similar or inconsistent entries. You also compare pre- and post-conversion state and county data to verify data integrity. Otherwise, the conversion could make state and county data unusable.

Structured Reports

Structured reports are run following conversion and reviewed to verify that they are in the desired form. For more information on structured reports, see *QAD Financials User Guide*.

Balance Sheet (25.15.5.4)

The Balance Sheet Report runs based on report structures implemented using the Budget function. The system constructs the balance sheet based on the accounts specified in the report structure. All other accounts are excluded.

Run the Balance Sheet report and check the resulting report for the correct structure.

Income Statement (25.15.5.5)

The Income Statement Report runs based on report structures implemented using the Budget function. The system constructs the income statement based on the accounts you specify in the report structure. All other accounts are excluded.

Run the Income Statement Report and check the resulting report for the correct structure.

Invoice Status Codes (36.1.11)

The conversion only creates three invoice status codes. Consider the number of codes needed for receiver and financial matching. Decide at what point in the process invoices are deemed approved, released for payment, and so on. Set up new status codes and assign them to suppliers (and customers) as appropriate.

Run the Invoice Status Code report and check the resulting report for the correct structure. Rename or replace the codes the conversion provides.

Daybooks

Daybooks are mandatory with the new Financials.

Daybook

1. Decide if any further reporting granularity is required.
2. Add new daybooks as necessary and assign to transaction types using Default Daybook Maintenance.
3. Optionally, delete or deactivate any unused daybooks.

Daybook Sets

The conversion creates two Daybook Sets: an AP Daybook set and an AR Daybook set. You can optionally use additional daybook sets to facilitate multiple invoice/credit note number ranges. Daybook sets can be defined for an entire domain or for individual sites. In the latter case, you assign the daybook sets to the desired sites.

If the source database has VAT tax registers and they are in use, the conversion creates a daybook set for each domain's VAT tax register/site combination. These daybook sets are in addition to the AP Daybook and AR Daybook sets the conversion creates.

Security

For more information on security, see *QAD Security and Controls User Guide*.

Roles and Permissions

Set up notification roles, including e-mail notification, and associated users for customers, suppliers, end users, and engineers. The roles and permissions used to grant access by role are similar to the roles and permissions in User Group Maintenance and Menu Security Maintenance in older versions.

Update Domain/Entity/User

Grant access by user to entities and domains.

Tax Periods

If the Create Entity Tax Periods parameter was set in the Conversion Parameters utility, the Tax Periods were created from the GL Periods during the conversion. However, if this parameter was not set, you must create Tax Periods manually within the application.

Create tax periods from the GL calendar. Otherwise, skip to "Reporting Periods".

Reporting Periods

Reporting periods are required when the new Financials budgeting is used. If GL Report Writer (GLRW) and budgets are used, reporting periods are not required. You must set up reporting periods to run structured reports.

If the new Financials budgeting is not used, skip to "Profiles".

Report periods mark a specific time span for producing budget reports. They are independent of GL periods and tax periods, and can span multiple GL periods across multiple entities.

Set up reporting periods.

Profiles

The conversion creates the required profiles for each element in a GL transaction. Review these profiles to verify that they are linked to the correct shared set.

Additionally, review the account codes for the specialized account profiles to ensure that the correct default sub-account, cost center, and project codes (as well as their associated linked shared sets) are set for each specialized account.

Configure Daemons

The History and Balance daemons are mandatory for correct system operation. You also review the budget and replication daemons for correct configuration. For more information on system daemons, see *QAD System Administration User Guide*.

Optional Post-conversion Set Up

Accounting Layers

The conversion creates transient and official layers. Add desired additional layers such as adjustments, internal reporting, and so on.

Cash Groups

Cash groups are used to group GL accounts for cash flow reporting purposes. Define new cash groups for petty cash (as opposed to the bank accounts for AR, AP, and Payroll).

Assign the cash groups to the required GL account codes.

Report Structures

Define any desired new reports. For more information on report structures, see *QAD Financials User Guide*.

Taxes

QAD Enterprise Edition uses a concept called Tax Groups. With tax groups (and boxes), you can set up tax reporting. During normal transaction processing, the application populates the tax transaction with the tax group defined against the tax rates. For more information on taxes, see *QAD Global Tax Management User Guide*.

Tax Codes

Define any new tax codes required to take advantage of new functionality such as Suspended/Delayed Taxes.

Tax Boxes

Tax boxes contain the individual elements of a transaction (for example, Tax Amount or Tax Base Amount, as they are reported in official returns to the authorities).

Tax Groups

A tax group contains one or more tax boxes.

Update Tax Rates

Add any new tax groups defined to the applicable tax rate.

Posting Tax Group Update

The conversion creates one default tax group. This default is assigned to transactions until the required setup is defined.

This utility updates existing tax transactions with new tax groups added to a tax rate. The updating allows transaction processing to start without the definition of tax groups/boxes.

Supplementary Analysis Fields

Define any SAF codes and assign to accounts, sub-accounts, and cost centers if this level of analysis is desired.

Customer Credit Checking

QAD Enterprise Edition significantly enhances customer credit checking. To take full advantage of all the new capabilities in this area, review the credit checking parameters to ensure that they are set according to the desired behavior for each customer.

Customer/Supplier Payment Statuses

Review and modify or create additional payment statuses for customers and suppliers.

Customer/Supplier Control Accounts

Review and modify or create additional control accounts for customers and suppliers.

Additional Profiles

Review and modify any new profiles set up for customer and supplier credit notes and prepayments.

Chart of Account (COA) Mask

Review and modify or create additional COA Mask records. Additional COA Mask records could be required for GL account codes the conversion creates when those accounts fall outside of current GL validation ranges. Examples of accounts outside of validation ranges are replacement accounts created within the GL Account Type Utility or accounts defined within the Conversion Parameters Utility. For more information on COA Mask records, see *QAD Financials User Guide*.

Chapter 5

Upgrading QAD Enterprise Edition into a New Environment

This section describes how to migrate and upgrade the components of an existing QAD Enterprise Edition environment into a new environment.

Overview

This section provides detailed instructions on how to migrate and upgrade an existing QAD Enterprise Edition environment into a new QAD Enterprise Edition 2022 environment. See Chapter 6 on page 81 for instructions on upgrading an existing QAD Enterprise Edition environment to QAD Enterprise Edition 2022.

At a high level, the process required involves:

- Creating a new YAB-installed EE environment
- Restoring the previous EE release databases (mfg, admin, help) from the existing environment into the newly installed EE environment
- Executing the EE upgrade process

The EE upgrade process in YAB uses the existing conversion routines from previous EE releases, and follows a similar sequence of steps. At a high level these steps are:

- Determining the source database version
- Determining the appropriate list of conversion routines to be executed
- Deleting system data records from both the qaddb and qadadm databases
- Upgrading the database schema and structure appropriately for qaddb, qadadm and qadhlp
- Executing the conversion routines for the qaddb and qadadm databases

Important Only use the following upgrade instructions to upgrade an existing Enterprise Edition installation to the latest release. Do not use this option if your current environment is not Enterprise Edition.

Upgrading Enterprise Edition into a New Environment

A QAD Enterprise Edition upgrade consists of the following tasks:

- Back Up Source Environment Databases
- Install QAD Enterprise Edition
- Restore the Source EE Databases
- Configure the Upgrade (Optional)
- Prepare Adaptive UX Environments
- Execute the Conversion Process
- Complete the Upgrade
- Migrate Process Map Changes
- Migrate Financial Attachments
- Update Custom Programs
- Migrate Custom Programs
- Migrate Custom System Data
- Post-upgrade Configuration
- Update the Patch Level

Back Up Source Environment Databases

Back up the existing environment database before starting the upgrade.

1 Shut down all of the processes associated with the environment being backed up, including all Progress processes and Tomcat.

- For YAB environments, enter `yab stop.`
- For QDT environments:

  a Stop the QAD Enterprise Edition environment by running the following script:
  `<QDT_install_directory>/envs/<environment_name>/scripts/stopenv.<environment_name>`

  b Shut down the Progress Admin Server.

  c Shut down Tomcat.

2 Back up the Progress databases, including all of the empty databases associated with the environment. Refer to the Progress documentation at www.progress.com/support for Progress backup procedures.

Note Only the mfg, admin and help databases are required for the EE upgrade process.

Install QAD Enterprise Edition

The first step in the upgrade process is to install the latest version of QAD Enterprise Edition. For detailed information about QAD Enterprise Edition installation, see *QAD Enterprise Edition Installation Guide*.

Important When installing QAD Enterprise Edition, do not install demo data. If you have already installed demo data, you must remove it. See the "Using Demo Data" section of the *QAD Administration and Configuration Guide* for YAB 1.15 on the QAD Document Library.

Restore the Source EE Databases

Once the 2022EE installation has completed successfully, remove the newly installed databases and replace them with the databases from the source environment.

1. Stop the 2022EE environment with the following command:

   ```
   > yab stop
   ```

2. Remove the newly installed 2022EE databases and their structure files:

   a. Change to the databases directory.

      ```
      > cd /<application-directory>/databases
      ```

   b. Remove the mfgdb database.

      ```
      > prodel mfgdb
      ```

   c. Remove the associated structure files

      ```
      > rm mfgdb.st
      ```

   d. Repeat steps b and c and remove the admin (admdb) and help (hlpdb) databases.

3. Restore the source EE databases from the source environment:

   ```
   > prorest mfgdb /<back-up-dir>/<mfgdb-back-up-name>.bkup
   ```

   Repeat this step for the admin (admdb) and help (hlpdb) databases.

   Once the databases have been restored, ensure that the file system permissions are consistent with the original database files, including any fixed-extents-related files. After restoring, the database files must have the same user and group permissions as the original files.

   The codepage of the source databases must match that of the 2022EE environment. The default for 2022EE is UTF-8.

   Note An additional Progress database conversion may be required if the source and target environments do not have the same Progress version. This is typically required only when there is a major version change, such as from OE 10.x to OE 11.x.

   Note Restoring a source EE database into an environment that has enabled secure configuration may result in the following error during the conversion.

   ```
   -P and -U startup parameters require an authentication enabled domain
   (16395)
   ```

   To resolve this issue, execute the following command for each source database before running an update, where `INSTANCE` is the database name, such as `qaddb` or `qadadm`.

   ```
   yab -clean database-INSTANCE-security-update
   ```

Note Any custom schema may be lost during the EE upgrade if the schema is not configured in YAB. For more details on how to manage schema in YAB, see the **Schema Changes** section of the *QAD Adaptive ERP Configuration and Administration Guide*.

Configure the Upgrade (Optional)

The `conversion-ee-upgrade` process has a number of optional configuration properties that are detailed in Table 5.1. These properties have default values and do not need to be explicitly set.

Table 5.1
Optional Configuration Values

Properties	Default	Description
conv-ui.ACDATE	None	ECommerce Turnaround data last accessed date. If blank the conversion uses the current date; that is, today.
conv-ui.ACUSER	mfg	ECommerce Turnaround data last accessed User ID. A value is required
conv-ui.CRDATE	None	ECommerce Turnaround data creation date. If blank the conversion uses the current date; that is, today.
conv-ui.CRUSER	mfg	ECommerce Turnaround data creation UserID. A value is required.
conv-ui.uiattachdir	${webapp.homeserver.dir}/configurations/${environment.id}/storage/attachments	The default location for Financial UI Attachments.

To view more detailed information for each configuration setting, use the yab help command. For more information regarding making configuration changes, refer to the *QAD Adaptive ERP Configuration and Administration Guide*.

Prepare Adaptive UX Environments

If you are executing the conversion on an Adaptive ERP environment that includes Adaptive UX, you must perform an additional step before executing the conversion process. If your environment does not include Adaptive UX, skip to the next step, Execute the Conversion Process.

Reset the auto-disabled Adaptive UX conversion. Enter:

```
> rm <appdir>/build/work/system/process-ignore
```

Execute the Conversion Process

Start the conversion with the following command:

```
> yab conversion-ee-upgrade
```

After the conversion completes, you should validate that it was successful by following the steps in Appendix H, “Validate the Conversion Process (conversion-ee-upgrade),” on page 203.

Note From yab-ee-app-1.18, the conversion-ee-upgrade process requires mandatory input parameters. For more details, see yab help conversion-ee-upgrade.

Complete the Upgrade

Once the conversion has completed successfully, reload the system data with the following command:

```
> yab -clean update
```

Rerun the Conversion Execution

The `conversion-ee-upgrade` process records state information to ensure that already-executed processes are not re-executed. If you want to rerun the conversion from the beginning, run the `conversion-ee-upgrade-reset` process and then the `conversion-ee-upgrade` process.

```
> yab conversion-ee-upgrade-reset
> yab conversion-ee-upgrade
```

Migrate Process Map Changes

If you have customized the standard process maps in the source environment, you must manually migrate the changes to the new environment.

1. Locate the `process-config.xml` file in the source environment.

   ```
   > ../conf/process-config.xml
   ```

2. Migrate the changes to the `process-config.xml` file in the new environment in the following location:

   ```
   > ../<application-directory/servers/tomcat/webapps/pronav/WEB-INF/conf/process-config.xml
   ```

Migrate Financial Attachments

Beginning with QAD 2011 Enterprise Edition, financial attachments that were previously stored in the database are now stored in the file system with all other attachments. Therefore, if your source environment is QAD 2011 Enterprise Edition or above, you must migrate the contents of your attachments directory to the newly installed attachments directory.

Update Custom Programs

In QAD 2010.1 Enterprise Edition and earlier, all of the product source files were located under `<qad_install_directory>` in the `xrc` directory (or `src` if you purchased the full source code). Beginning with QAD 2011 Enterprise Edition, source files are located in `xrc/us` in a series of two-letter subdirectories with names that mirror the R-code subdirectories.

You must update custom programs that reference QAD include files and/or subprocedures to prefix these references with the two-letter subdirectory structure. Nearly all references in custom programs to QAD source files require a `us/<2_letter_subdirectory>/` prefix.

For example, custom program `xxabcdrp.p` is a menu-level report that uses QAD's `mfdtitle.i`. Before QAD 2011 Enterprise Edition, this line of code would have appeared as:

```
{mfdtitle.i}
```

After implementing QAD 2011 Enterprise Edition, this same line of code must read as:

```
{us/mf/mfdtitle.i}
```

Some include files are located in `xrc/us/bbi` rather than in their two-letter subdirectory. Check the `xrc/us/bbi` directory for the list of those include files. Following implementation of QAD 2011 Enterprise Edition, references in custom programs to include files in that subdirectory require a prefix of us/bbi/. For example:

```
{us/bbi/pxmsg.i &MSGNUM=4874 &ERRORLEVEL=1}
```

Migrate Custom Programs

Once the customizations have been updated, they must be correctly located within the new directory structure. For example, Operational Customization must be located here:

```
> ../<install-directory>/customizations/mfg
```

For more information, see the *QAD Adaptive ERP Configuration and Administration Guide*.

Once correctly located, the customizations should be compiled using the following command:

```
>yab update
```

Migrate Custom System Data

Upgrading does not convert or upgrade custom data changes, but the upgrade process does delete all records from a number of tables, defined in `inpempty.lst`. QAD-owned system data is subsequently reloaded during the upgrade, but custom data is not.

If you added custom data to any of the following default system data tables, you must dump the customized records before beginning an upgrade. If you have not made any customizations, skip this section.

- Custom menus (mnd_det, mnt_det) and messages (msg_mstr)

  Note You must reassign Roles after loading the custom menus. See the *QAD Security Administration Guide* for information on defining Role membership.
- Custom labels (lbl_mstr, lbld_det)
- Custom language detail (lng_det)
- Custom field help lookup (flhd_det, flhm_mstr)
- Custom reports, browses (use the import/export function in report resource export and browse maintenance), and favorites used in the QAD .NET UI (collections and metrics stored in Tomcat under storage directory in qadhome)

Note Table names are for information purposes only. The list of tables may be incomplete and is subject to change.

After dumping your custom data and completing the upgrade, incorporate your customizations in your target version environment. For more information, see the *QAD Adaptive ERP Configuration and Administration Guide*.

Post-upgrade Configuration

Following upgrade, Enterprise Edition or its components may require configuration. For more information on Enterprise Edition configuration, see *QAD Enterprise Edition Installation Guide*.

Update the Patch Level

The QAD Enterprise Edition patch level may require updating. For more information, see the *QAD Adaptive ERP Configuration and Administration Guide*.

Chapter 6

Upgrading an Existing QAD Enterprise Edition Environment

This chapter describes how to upgrade an existing QAD Enterprise Edition environment. This approach is only supported for QAD Enterprise Edition environments where the source environment was created using YAB, which applies to QAD Enterprise Edition 2016 and later.

Overview

The following procedure describes upgrading an existing 2016 or later QAD Enterprise Edition environment to 2022.

Note If the source environment contains any QAD add-on products, you must check their compatibility with QAD Enterprise Edition 2022 before performing the upgrade. If any add-on products need to be updated as part of this upgrade, refer to the individual add-on product guides as reference. QAD recommends that all QAD add-on products be installed using a delayed install to allow all products to be configured simultaneously when executing the final `yab update`.

Upgrading QAD Enterprise Edition 2016 or Later

Follow these steps to upgrade QAD Enterprise Edition 2016 or later to 2022.

Determine if you are downloading the installation zip file or mounting the install image from DVD and follow the appropriate steps to install a new instance of Enterprise Edition.

Installing with a Downloaded Zip File

1. Download the zip file from QAD Fulfillment.
2. Unzip the file into a subdirectory.

   ```
   unzip qad-ee-VERSION.zip -d <2022EE directory>
   ```
3. Change to the directory where the zip file was extracted.
4. Continue with Installing QAD Enterprise Edition.

Installing from DVD

1. Create the directory where the image will be mounted.

   ```
   mkdir directory
   ```
2. Mount the DVD.

   ```
   mount qad-ee-VERSION.zip <2022EE directory>
   ```
3. Change to the directory where the image was mounted.

Installing QAD Enterprise Edition

1. Stop the environment.

   ```
   yab stop
   ```
2. If your system includes EAM and you are upgrading to EAM 2017.1, you must perform a delayed installation of EAM 2017.1.

   ```
   yab -install-update:false install eam-yab-compatible-VERSION.zip
   ```

   If your system includes EAM 2016.1 or EAM 2017, you must install the EAM patch specific to your EAM version before proceeding. Download the patch from QAD and then execute the following command:

   - EAM 2016.1: `yab install eam-patch-13.1.2.x.zip`
   - EAM 2017: `yab install eam-patch-13.2.1.x.zip`
3. Perform a delayed installation of Enterprise Edition 2022, which should take about 10 minutes.

   ```
   yab -install-update:false install <2022EE directory>
   ```

   See the *QAD Enterprise Edition Installation Guide* for detailed instructions.

Execute the Conversion Process

Start the conversion with the following command:

```
yab conversion-ee-upgrade
```

After the conversion completes, you should validate that it was successful by following the steps in Appendix H, “Validate the Conversion Process (conversion-ee-upgrade),” on page 203.

Note From yab-ee-app-1.18, the conversion-ee-upgrade process requires mandatory input parameters. For more details, see yab help conversion-ee-upgrade.

Complete the Upgrade

To complete the upgrade, reload the system data with the following commands:

1. Update the environment scripts.

   ```
   yab script-update
   ```

2. Reload qaddb data.

   ```
   yab -clean database-qaddb-data-update
   ```

3. Reload qadadm data.

   ```
   yab -clean database-qadadm-data-update
   ```

4. Update YAB.

   ```
   yab update
   ```

Appendix A

GTM Conversions

This appendix describes GTM conversion considerations.

GTM Conversions Summary

Before eB, MFG/PRO supported four tax processing systems:

- Value-added tax (VAT)
- United States (US) taxes
- Canadian taxes
- Global Tax Management (GTM)

See “Converting to GTM From No Taxes” on page 118.

Note Some companies have not implemented a tax system because they use an external tax package or have an alternate method for tracking taxes. In this case, some preparation is still required to convert to GTM and continue without a tax system.

Of the four systems, GTM offers the most precise calculations and the greatest flexibility for calculating taxes for multiple countries. Therefore, the system no longer supports regional tax systems.

QAD supplies programs with GTM to automate most conversion tasks (Table A.1). These programs generate GTM codes and update existing eB2.1 and later records. They also create records required for tax reporting.

Table A.1
GTM Conversion Programs

Menu Number	Program
2.13.22.1	VAT to GTM–Setup
2.13.22.2	VAT to GTM–Masters
2.13.22.3	VAT to GTM–Transactions
2.13.22.5	USA to GTM–Setup
2.13.22.6	USA to GTM–Masters
2.13.22.7	USA to GTM–Transactions
2.13.22.9	CAN to GTM–Setup
2.13.22.10	CAN to GTM–Masters
2.13.22.11	CAN to GTM–Transactions

Pre-conversion Planning

To save time and reduce the likelihood of errors, address the following issues before starting the GTM conversion process:

- Timing

  Perform the conversion any time in transaction processing. You do not have to close open transactions or post transactions to the general ledger beforehand. However, for a clearer division of reporting, consider converting at the beginning of a new financial period.

  To prevent record contention conflicts, only run the conversion programs when no one else is using the system.

- Records to Convert

  Determine the range of records to convert. Master records such as customers and items are converted first and then the transaction records. Records are converted in the order in which their selection options display on the conversion screen. Finally, transactions that are prerequisites for other transactions are converted first. For example, purchase order receipts are converted before their respective vouchers.

  The conversion programs select records by number rather than the creation date or effective date. To convert records for a specific date range, specify the first record number for the starting date and/or the last record number for the ending date.

  The conversion programs do not cross-check the record selected for conversion. For example, for accounts payable, they do not verify that selected payment records are the ones associated with the selected voucher records.

- Code Naming Conventions

  Each conversion has default naming conventions for GTM tax classes, tax zones, and tax environments. Review the conventions and decide if they are what you want.

- Integration of GTM Enhancements

  The objective of the conversion is to move your existing tax processing configuration into GTM. You must complete the conversion before you can incorporate new GTM features. There are two reasons for this requirement. First, your current configuration does not have the data to support these features. Second, some of the conversion subprograms expect to encounter specific data values. They do not run correctly when you prematurely change these values.

- Custom Programming

  For some situations, custom programming is required. An example of such a situation is the need to merge two VAT class codes to one GTM tax class code without using GTM tax usage codes.

Practice running the conversion on a copy of your live database. This task allows you to identify issues in the existing records and familiarizes you with the conversion process.

Post-conversion Procedures

Once you complete the conversion, you must still exercise some control to ensure a clean division for pre- and post-GTM reporting:

- Handling of Closed Transactions

  Never reverse or delete closed transactions that were not included in the conversion once you start using GTM.

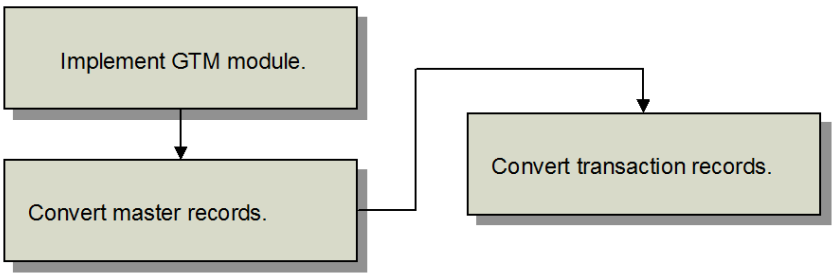
- Effective Date for Tax Reports

  Print tax reports with a post-conversion tax date to exclude transactions you did not bring into GTM.

Converting VAT Taxes to GTM

This section describes conversion from the VAT system available in earlier versions to GTM. The VAT to GTM conversion translates VAT data to GTM equivalents and updates existing eB2.1 and later records. Figure A.1 summarizes the conversion workflow.

Fig. A.1
VAT to GTM Conversion Process



The workflow revolves around three sets of activities:

Implement GTM Module. Implementing GTM for VAT, you make planning decisions and then run a set up program. How this program works depends on whether a country is a member of the European Community.

Convert Master Records. Run a program that populates database tables for customer and supplier addresses, items, product lines, and other master records with GTM data values.

Convert Transaction Records. Run a second program that populates transaction records. All transactions subject to tax are affected, including sales, purchasing, accounts payable, accounts receivable, and service/support management.

Important After each of these activities, review the corresponding reports and audit trails. Mistakes can be pervasive and costly.

Table A.2 lists the eB2.1 and later programs used during the conversion.

Table A.2
eB2.1 and later Programs Used to Convert VAT to GTM

Activity	eB2.1 and Later Programs
Implementing GTM	Country Code Maintenance (2.14.1)
	VAT to GTM–Setup (2.13.22.1)
Converting master records	VAT to GTM–Masters (2.13.22.2)
Converting transaction records	VAT to GTM–Transactions (2.13.22.3)

Implementing GTM

An automated set up program can create most of the codes required to implement GTM, based on how your VAT taxes are defined. Before executing this program, learn the options it provides and the default logic it uses.

Country Codes and Tax Environments

The first step in implementing VAT taxes in GTM is to ensure that all countries are defined in Country Code Maintenance (2.14.1). Countries that are part of the European Community must have the EC field set to Yes. This step is important because two options are available during the automated setup:

1. You can create tax zones and tax environments with countries summing into one of two predefined tax zones: union and non-union. To do this task, specify a code representing the union. By default, this code is EU. The system then creates two tax zones—EU and NON-EU. These zones correspond to special country code values of ~1 (EU) and ~0 (NON-EU).

   This approach results in a minimum number of tax environments:

   - One environment for each country defined in the country master.
   - IN-EU for transactions within the union but outside the borders of one country
   - FROM-EU for transactions between a member of the union and a nonmember
   - TO-EU for transactions from a nonmember to a member of the union
   - NON-EU for transactions between two countries that are both nonmembers of the union

2. You can create tax zones and environments based on each defined country without summing into a union. This option results in the maximum number of tax environments - one for each combination of ship-from and ship-to countries. If, for example, you do business with 10 countries, 100 tax environments are created.

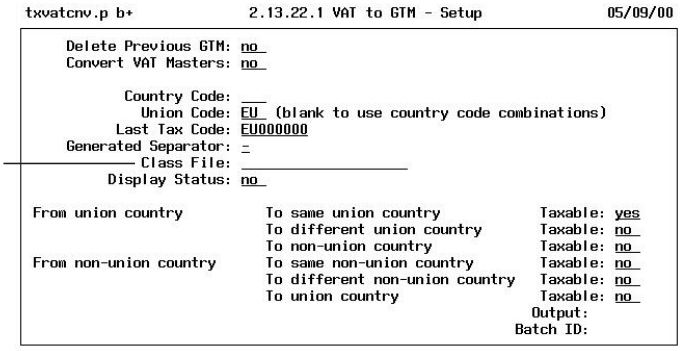
   If you choose this option, you do not need to specify a union code. The set up program does not create the EU and NON-EU zones or the ~1 and ~0 countries.

Defining Custom Tax Class and Usage Codes

By default, the conversion setup generates tax classes that correspond to your VAT classes, without associated GTM tax usage codes. You can override this behavior by creating your own map for the setup program and specifying it in the Class File field. The programs that convert master data and transaction records also reference the same class file.

Fig. A.2
Class File in VAT to GTM–Setup (2.13.22.1)

Class file for custom Tax Class and Tax Usage codes



Create a class file if:

- Your company plans to change tax class codes during the conversion.

- Within a tax class, a company can be taxed based on its nature of operation or the way it intends to use an item. Tax usage codes identify these conditions in GTM.

The class file is an ASCII file with text strings in the following format:

```
"Current VAT Class" "GTM Tax Class" "GTM AP Tax Usage" "GTM AR Tax Usage"
```

GTM tax classes are a maximum of three characters, and tax usage codes are eight characters. A null string (" " or "") represents an unused optional value.

The class file can have any name or extension. However, commas instead of blank characters separate code values in .csv files. Place the file in the home directory for the Progress session. A .csv file is a Windows comma-separated value file format that saves recorded values in a spreadsheet.

The class file accommodates companies that use different tax class and/or tax usage codes for AP and AR processing. If you use one set of codes for both kinds of tax processing, simply specify the same usage code for both.

Example Your current VAT classes are 1 and 2. You want to map VAT class 1 to GTM tax class A, tax usage code FOOD, and VAT class 2 to GTM tax class B, tax usage DRUG.

```
"1" "A" "FOOD" "FOOD"
"2" "B" "DRUG" "DRUG"
```

AP and AR usage codes are applied differently during the conversion to master records and transactions.

Table A.3
AP and AR Usage Codes

Usage Code	Applied to...
AP Usage Codes	The master conversion applies AP usage codes to supplier records. The transaction conversion uses them to update purchasing and accounts payable records.
AR Usage Codes	The master conversion applies AR usage codes to customers, warranty types, and contract types. The transaction conversion uses them to update sales, accounts receivable, and service/support management transactions.

Processing Logic

VAT to GTM–Setup creates records as described in Table A.4.

Table A.4
New GTM Records

Type of Record	Explanation
Tax zones	Based on the value of Union Code, setup builds the tax zone hierarchy for EU and non-EU countries in your current system or tax zones for all countries.
Tax types	Setup creates a tax type of VAT and NON-TAX.
Tax environments	Based on the value of Union Code, setup generates sums-into tax environments or environments for all ship-to/ship-from country combinations.
Tax rates	Based on VAT rates, setup generates tax rates for the tax jurisdictions and percentages used in your current system, as well as a nontaxable tax rate.
Tax classes	By default, setup generates tax classes based on your current VAT classes and a NOT-TAX class for nontaxable transactions. To create different classes, define a class file (see "Defining Custom Tax Class and Usage Codes" on page 89).

Type of Record	Explanation
Tax usages	By default, setup does not generate tax usages. However, you can create these usages with a class file.
Country code	The setup generates a record for the default country code you specify when you run the setup. If you enter a union code, setup creates `~1` and `~0` country codes.
Company addresses and address list types	In GTM, company sites require a corresponding company address record because taxes are calculated using address, not site. The setup verifies that each company site has an address record and creates any missing ones, along with any needed address list type records. The setup also creates a `~taxes` address record to provide a default tax address whenever a transaction is missing a company site code.

GTM Control Settings

Setup defines Global Tax Management Control (2.13.24) as described in Table A.5.

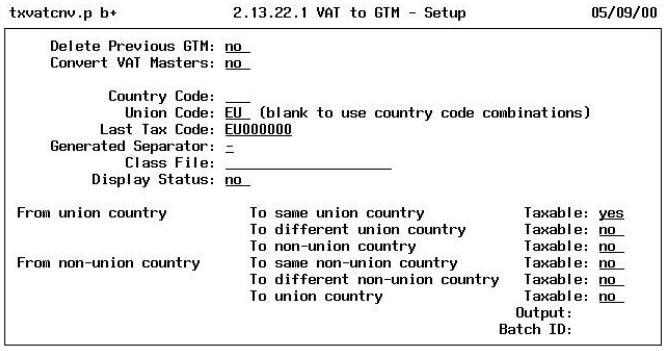
Table A.5
Updates to GTM Control Settings

Field	Explanation
Country Code	Value specified in VAT to GTM–Setup
Tax Method	01
Tax-By-Line	No
Accrue Tax at Receipt	No
Discount Tax at Invoice	Same setting as in VAT Control (2.15.2.24)
Discount Tax at Payment	Same setting as in VAT Control (2.15.2.24)
Last Tax Code	Value specified in VAT to GTM–Setup

VAT to GTM–Setup

Based on your implementation decisions, use VAT to GTM–Setup (2.13.22.1) to set up GTM for VAT tax processing.

Fig. A.3
VAT to GTM–Setup (2.13.22.1)



Delete Previous GTM. This option determines whether the setup deletes previously created GTM records from the database. If you select this option, the setup deletes tax zones, tax types, tax environments, transaction tax details, and other GTM records.

- Enter No if you have not yet converted your database to GTM.

- Enter Yes to clean up the database if it contains GTM records from unsuccessful conversion or installation attempts.

Convert VAT Masters. This option determines whether setup generates GTM records based on country codes and VAT classes.

- Enter Yes to create GTM records corresponding to VAT classes.
- Enter No if you only want to delete previous GTM records and do not want the setup program to generate new GTM records.

Country Code. Enter the default country code for the GTM control program.

Union Code. Enter a three-character code (default is EU) representing the European Union if you want countries to sum into a union and non-union zone. Leave this code blank to create tax environments for each combination of ship-to and ship-from countries.

Last Tax Code. Enter a value to update the corresponding field in the Global Tax Management Control. In GTM, tax codes identify individual tax rates. Codes are generated sequentially based on the value of Last Tax Code in the GTM control program.

The default Last Tax Code is the union code followed by zeros. For example, for union code EU, the default Last Tax Code is EU000000. This value is recommended if you specified a value in Union Code.

If you are not summing into a union code, enter the default country code followed by zeros.

Generated Separator. Enter a character to use as a separator in system-generated tax zones and environments. Using a separator can improve the readability of the component elements of these codes.

The default separator is the dash (–), but you can enter any character. A sample GTM code that uses the dash separator is PAR–FR for Paris, France. If you do not want to use separators, enter blank. However, you cannot use blank as a separator character.

Note The system-generated nontaxable tax type is NON-TAX, regardless of the separator you specify.

Class File. To provide custom mapping of VAT classes to GTM classes and usage codes, specify an ASCII file with conversion information.

See “Defining Custom Tax Class and Usage Codes” on page 89.

Display Status. This setting determines whether the system displays status messages online during the conversion. These messages list database tables and their indexes as they are converted. If you select this option, messages display on the screen and the printed report.

Taxable. Enter the appropriate values for the six possible combinations of transactions between EC countries and non-EC countries. Yes indicates that the transaction is taxable.

Converting Master Records

Once you finish implementing VAT processing in GTM, the next activity is to update tax settings in the following master records:

- Suppliers
- Customers
- Product Lines

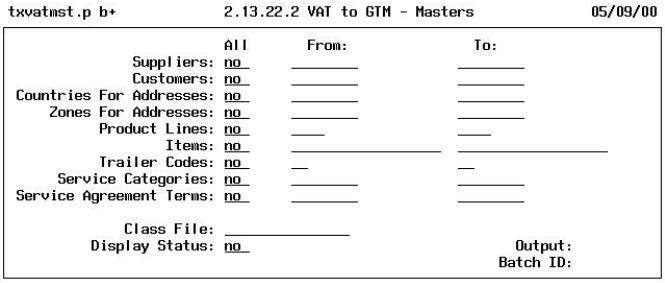
- Items
- Trailer Codes
- Service Categories
- Service Agreement Terms

GTM has additional fields and could require new values for existing fields.

VAT to GTM–Masters

To convert master records, run VAT to GTM–Masters (2.13.22.2).

Fig. A.4
VAT to GTM– Masters (2.13.22.2)



In addition to updating the tax settings in the master records, this program assigns tax zone codes to supplier, customer, and company address records. For verification of changes, the program generates an audit trail.

For each type of record, you can convert all records, a range of records, or individual records. The program converts records in the same order as the options on the screen. For separate audit trails, run the report separately for each type of record.

If you created a class file during the setup step, specify its name in the Class File field. Display Status, Output, and Batch ID are the same as in the setup program.

See “Defining Custom Tax Class and Usage Codes” on page 89.

Master Conversion Audit Trail

The master conversion prints a report of changed records. The format varies depending on the records included in the conversion. For each group of converted records, the report shows the record number and name and the before and after tax information. This tax information includes country code, tax zone code, taxable status, whether tax is included in item amounts, tax class, and tax usage.

Groups of converted records print in the same order as the screen selection criteria, and a page break separates each group. Warning and error messages identify potential conversion issues.

If you specified a class file, the report prints the VAT class and the corresponding GTM tax class and tax usage, if any.

Figure A.5 and Figure A.6 show representative audit trail formats.

Fig. A.5
Customer Audit Trail

```
Processing: Customers

Address Name                                  Taxable Tax In TxC TaxUsage
-------- ----------------------------        ------- ------ --- --------
10000001 Consolidated Industries Ltd. Before     No      No     E   1-P-MFG
                                      After      No      No     E   1-P-MFG
10000002 Office Automation B.V.       Before     Yes     No     H   1-P-MFG
                                      After      Yes     No     H   1-P-MFG
10000003 MMB Verkehrssysteme GmbH     Before     Yes     No     G   1-P-MFG
                                      After      Yes     No     G   1-P-MFG
```

Fig. A.6
Countries for Addresses Audit Trail

```
Processing: Countries For Addresses

Address Name                                 Ctry Country
-------- ----------------------------        ---- --------------------------
10000001 Consolidated Industries Ltd. Before      United Kingdom
                                      After  UK   United Kingdom
10000002 Office Automation B.V.       Before     Netherlands
                                      After  NL   Netherlands
10000003 MMB Verkehrssysteme GmbH     Before     Germany
                                      After  D    Germany
```

Troubleshooting the Master Conversion

The error messages in the audit trail identify conditions to analyze and address before you convert transactions. Table A.6 lists some common errors along with explanations. Restore the database from backup before making corrections.

Warning Do not proceed to the transaction conversion until the master conversion audit trail is free of errors.

Table A.6
Troubleshooting the Master Conversion

Error	Explanation
Tax class cannot be converted.	Class file does not contain a VAT class that matches the one in the master record.
Blank tax class not allowed.	VAT class is blank in the class file.
Tax class cannot exceed 3 characters (*xxx*).	VAT class in the class file is longer than three characters. Message shows the first three characters.
Tax class does not exist (*x*).	VAT class in the class file is not in the VAT master.
Tax class is not unique (*x*).	VAT class occurs in multiple places in the class file.
Tax class does not exist (*xxx*).	GTM tax class in the class file does not exist in the GTM tax class master.
Tax usage cannot exceed 8 characters (*xxxxxxxx*).	GTM tax usage in the class file is longer than eight characters. Message shows the first eight characters.
Tax usage does not exist (*xxxxxxxx*).	GTM tax usage in the class file does not exist in the GTM tax usage master.
Tax class/tax usage combination is not unique (*xxx xxxxxxxx*).	GTM tax class and tax usage combination occurs in multiple places in the class file.

Note *x*, *xxx*, and *xxxxxxxx* are placeholders for the actual codes displayed in the error message.

How the Conversion Changes Master Records

The following is a technical description of how the master conversion updates the database.

The menu-level program for VAT to GTM–Masters is `txvatmst.p`. This program calls subprograms that set the GTM tax values in the individual database tables. These programs can set the existing VAT class value or retrieve an alternate value from a class file.

Table A.7 lists the affected database tables and summarizes the changes.

Table A.7
Changes to Master Records

Table	Summary of Changes
Address Master (ad_mstr)	In supplier records, `txvatvd.p` sets ad_taxable from vd_taxable. It also sets ad_taxc and ad_tax_usage from ad_taxc or from the class file with AP usage, if any. In customer records, `txvatcm.p` sets ad_taxable from cm_taxable and ad_tax_in from cm_tax_in. It also sets cm_taxc, ad_taxc, and ad_tax_usage from cm_taxc and from the class file with AR tax usage, if any. In all address records, `txvatct.p` sets ad_ctry from ad_country and the other way around. `txvatzn.p` calls `txtxzget.p` to set ad_tax_zone.
Service Category Master (fsc_mstr)	`txvatfsc.p` sets fsc_taxc from fsc_taxc or from the AR tax usage, if any.
Product Line Master (pl_mstr)	`txvatpl.p` sets pl_taxc from pl_taxc or from the class file if any.
Item Master (pt_mstr)	`txvatpt.p` sets pt_taxc from pt_taxc or from the class file if any.
Service Agreement Terms and Conditions Master (sv_mstr)	`txvatsv.p` sets sv_taxc from sv_taxc or from the class file with AR tax usage, if any.
Trailer Master (trl_mstr)	`txvattrl.p` sets trl_taxc from trl_taxc or from the class file, if any.

Converting Transaction Records

Once you finish converting master records, you can convert the following transaction records:

- Purchase orders and receipts
- Accounts payable vouchers and payments
- Service contracts, calls, and return material authorizations
- Sales quotes and orders
- Accounts receivable memos, invoices, and payments

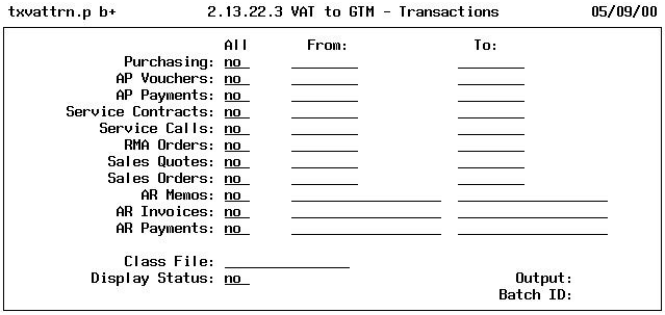
In GTM, every transaction subject to tax has a transaction tax detail record. This record stores the information used to calculate tax. It also separates the tax into component elements such as recoverable and non-recoverable amounts.

Note If you already have GTM transaction records in the database when you perform transaction conversion, the conversion process updates them using the current default tax values. Values replaced in these records include tax environment, class, usage, and so on.

VAT to GTM–Transactions

To convert existing transaction records so they are accessible in GTM, run VAT to GTM–Transactions (2.13.22.3).

Fig. A.7
VAT to GTM– Transactions (2.13.22.3)



In addition to updating transactions, this program generates an audit trail for verification of changes.

You can convert all records, a range of records, or individual records. The program converts records in the same order they display on the screen.

Note In some cases, the record sequence is important. Purchasing transactions are converted before accounts payable vouchers and vouchers before payments. Accounts receivable memos and invoices are converted before payments.

If you created a class file during the setup step, specify its name in the Class File field. Display status, output, and Batch ID are the same as in the setup program.

Transaction Audit Trail

The transaction conversion prints a report of changed records. The format varies depending on the records included in the conversion. For each group of converted records, the report shows the transaction number and name and the before and after tax information for each line item. This tax information includes taxable status, tax environment, tax class, and tax usage. Groups of converted records print in the same order as the screen selection criteria with a page break separating each group.

Warning and error messages identify potential conversion issues. Messages that appear at the end of a transaction apply to the entire transaction; messages that appear between the Before and After line apply only to that line. If you specified a class file, the report prints the VAT class and the corresponding GTM tax class and tax usage, if any.

Figure A.8 and Figure A.9 show representative audit trail formats.

Fig. A.8
Purchasing Audit Trail

```
Processing: Purchasing

Order    Receiver  Ln         Tax TxC TaxUsage Tax Env  prh_tax_at
-------- -------- ---         --- --- -------- -------- ----------
01104533             Before  No
                     After   No  E            BE-NE
                   1 Before  Yes e
                     After   Yes E            BE-NE
         RC1290    1 Before      e                    E
                     After       E            BE-NE    Yes
```

Fig. A.9
Service Calls Audit Trail

```
Processing: Service Calls

Call ID  Call/SR  Line Record               Tax TxC TaxUsage Tax Env
-------- -------- ---- ----------           --- --- -------- ----------------
CA127                  Call       Before     No  0
                                  After      No  0   1-P-MFG  GER-NE
         CA127       1 Item       Before     No  0
                                  After      No  0   1-P-MFG  GER-NE
         CA127       1 Billing    Before     No  0
                                  After      No  0   1-P-MFG  GER-NE
```

Troubleshooting Transaction Conversion

The warning and error messages in the audit trail identify conditions you analyze and address before resuming live GTM processing. Table A.8 lists a common error along with an explanation. Before you make corrections, restore the database from backup.

Warning Do not resume live processing until the transaction conversion audit trail is free of errors.

In addition to examining the audit trail, review the Tax Detail by Transaction Report (2.13.15.3). This report shows the tax environments, tax types, and tax amounts for the converted records. Verify that the tax calculations produced the expected results.

Note Converted transactions could have minor differences in before/after tax amounts. The differences can occur because GTM uses a different calculation algorithm or rounding method than your current system. To synchronize the general ledger with the converted transactions, record adjusting entries.

Table A.8
Troubleshooting the Transaction Conversion

Error	Explanation
Detail tax environment must match header.	In accounts payable vouchers and accounts receivable debit/credit memos, the tax environment are the same in both the header and detail lines.

Warning Do not correct transaction records programmatically because doing so can cause additional problems.

To eliminate ambiguity, the audit trail shows before and after values for purchasing, accounts payable, and accounts receivable records by their Progress database field name, not their screen label. For example, the audit trail displays voucher line types in the vod_type column.

Use Table A.9 to Table A.13 to interpret audit trails for the transaction conversion. These tables summarize before and after tax values.

Note In the following tables, quotation marks indicate a value that cannot be translated.

Table A.9
VAT to GTM, Purchasing Transactions

Status	Tax System	pod_taxable	prh_tax_at
Taxable	VAT	Yes	VAT class
	GTM	Yes	"Yes"
Nontaxable	VAT	No	"0"[1]
	GTM	No	Blank

1. First VAT class with a zero percentage.

Table A.10
VAT to GTM, AP Voucher Receiver Lines

Status	Amt	Tax System	Tax	TxC	vod_type	vod_tax	vod_tax_at
Taxable	Item	VAT	No	VAT class	"R"	Blank	VAT class
		GTM	Yes	Tax class	"R"	Blank	"Yes"
	Tax	VAT	No	Blank	Blank	VAT class	Blank
		GTM	No	Blank	"T"	"t"	"No"
Nontaxable	Item	VAT	No	VAT class	"R"	Blank	VAT class
		GTM	No	Tax class	"R"	Blank	Blank
	Tax[1]	VAT	No	Blank	Blank	VAT class	Blank
		GTM	–	–	–	–	–

1. The conversion deletes VAT tax lines resulting from nontaxable amounts.

Table A.11
VAT to GTM, AP Voucher Memo Lines

Status	Amt	Tax System	Tax	TxC	vod_type	vod_tax	vod_tax_at
Taxable	Item	VAT	No	Blank	Blank	Blank	VAT class
		GTM	Yes	Tax class	Blank	Blank	"Yes"
	Tax	VAT	No	Blank	Blank	VAT class	Blank
		GTM	No	Blank	"T"	"t"	"No"
Nontaxable	Item	VAT	No	Blank	Blank	Blank	"0"[1]
		GTM	No	Tax class	Blank	Blank	"No"
	Tax	VAT	No	Blank	Blank	VAT class	Blank
		GTM	–	–	–	–	–

1. The conversion deletes VAT tax lines resulting from nontaxable amounts.

Table A.12
VAT to GTM, AR Invoices

Status	Amt	Tax System	TxC	ard_tax	ard_tax_at
Taxable	Item	VAT	Blank	VAT class	Blank
		GTM	Tax class	Blank	Tax class
	Tax	VAT	Blank	Blank	VAT class
		GTM	Blank	"t"	"No"
Nontaxable	Item	VAT	Blank	Blank	"0"[1]
		GTM	Tax class	Blank	Tax class
	Tax[2]	VAT	Blank	"0"	Blank
		GTM	–	–	–

1. First VAT class with a zero percentage.
2. The conversion deletes VAT tax lines resulting from nontaxable amounts.

Table A.13
VAT to GTM, AR DR/CR Memos

Status	Amt	Tax System	TxC	ard_tax	ard_tax_at
Taxable	Item	VAT	Blank	Blank	VAT class
		GTM	Tax class	Blank	"Yes"
	Tax	VAT	Blank	VAT class	Blank
		GTM	Blank	"t"	"No"
Nontaxable	Item	VAT	Blank	Blank	"0"
		GTM	Tax class	Blank	"No"
	Tax[1]	VAT	Blank	"0"[2]	Blank
		GTM	–	–	–

1. The conversion deletes VAT tax lines resulting from nontaxable amounts.
2. First VAT class with a zero percentage.

How the Conversion Changes Transaction Records

The following is a technical description of how the transaction conversion updates the database.

The menu-level program for VAT to GTM–Transactions is `txvattrn.p`. This program calls subprograms that set the GTM tax values in the individual database records. For all transactions, the conversion also generates corresponding tax detail records in the Tax Detail (tx2d_det) database table.

When setting the GTM tax class value, these programs can set the existing VAT class value or retrieve an alternate value from a class file.

Table A.14 lists the affected database tables and summarizes the changes.

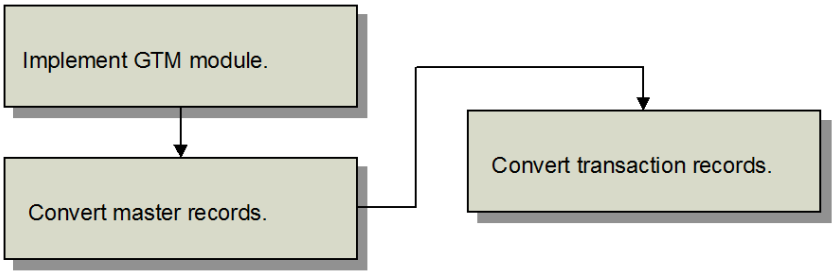
Table A.14
Changes to Transaction Records

Tables	Summary of Changes
Accounts Receivable Detail (ard_det)	For debit/credit memos, `txvatarm.p` sets ard_tax, ard_tax_at, ard_taxc, and ard_tax_usage from the class file with AR usage, if any. For invoices, `txvatari.p` sets ard_taxc and ard_tax_usage from the class file with AR tax usage, if any, and ard_tax and ard_tax_at. Duplicate records for unique keys are merged into one record.
Accounts Receivable Master (ar_mstr)	For debit/credit memos, `txvatarm.p` sets ar_tax_env.
Service/Support Call Master (ca_mstr)	`txvatca.p` sets ca_taxc and ca_tax_usage from ca_taxc or from the class file with AR tax usage, if any. It also sets ca_tax_env using `txtxeget.p`.
Invoice History Detail (idh_hist)	`txvatari.p` sets idh_taxc and idh_tax_usage from idh_taxc or from the class file with AR tax usage, if any. It also sets idh_tax_env using `txtxeget.p`.
Invoice History Master (ih_hist)	`txvatari.p` sets ih_taxc and ih_tax_usage from ih_taxc or from the class file, with AR tax usage, if any. It also sets ih_tax_env using `txtxeget.p`.
Service/Support Call Item Detail (itm_det)	If itm_prefix is CA and itm_type is any value except INV, `txvatca.p` sets itm_taxc and itm_tax_usage from itm_taxc or from the class file with AR usage, if any. It also sets itm_tax_env using `txtxeget.p`.
Purchase Order Detail (pod_det)	For inventory items, `txvatpo.p` sets pod_taxc and pod_tax_usage from pod_taxc or from the class file, with AP tax usage, if any. For memo items, `txvatpo.p` sets pod_taxc from po_taxc and pod_tax_usage from po_tax_usage.
Purchase Order Master (po_mstr)	`txvatpo.p` sets po_tax_pct[1], po_tax_pct[2], and po_tax_pct[3] to 0. It sets po_tax_usage from ad_tax_usage and po_taxc from ad_taxc. It also sets po_tax_env using `txtxeget.p`.
Purchase Order Receipt History (prh_hist)	For inventory items, `txvatpo.p` sets prh_taxc and prh_tax_usage from prh_taxc or from the class file, with AP tax usage, if any. For memo items, `txvatpo.p` sets prh_taxc from pod_taxc and prh_tax_usage from pod_tax_usage. For all items, `txvatpo.p` also sets prh_tax_at from pod_taxable.
Sales Quotation Detail (qod_det)	`txvatqo.p` sets qod_taxc and qod_tax_usage from qod_taxc or from the class file with AR usage, if any. If qo_taxable and qod_taxable are No and the quote is for an inventory item, `txvatqo.p` sets qod_taxc from pt_taxc. It also sets qod_tax_env.
Sales Quotation Master (qo_mstr)	`txvatqo.p` sets qo_tax_pct[1], qo_tax_pct[2], and qo_tax_pct[3] to 0. `txvatqo.p` also sets qo_taxc and qo_tax_usage from qo_taxc or from the class file with AR usage, if any. It also sets qo_tax_env.
Return Material Authorization Master (rma_mstr)	`txvatrma.p` sets rma_taxc from rma_taxc or from the class file with AR usage, if any.
Service Contract Detail (sad_det)	`txvatsc.p` sets sad_taxc and sad_tax_usage from sad_taxc or from the class file with AR usage, if any. `txvatsc.p` sets sad_tax_env from sa_site and sa_taxc using `txtxeget.p`.

Converting US Taxes to GTM

This section describes the conversion from MFG/PRO US taxes to GTM. The USA to GTM conversion process translates United States tax data to GTM equivalents and updates existing eB2.1 and later records. Figure A.10 summarizes the conversion workflow.

Fig. A.10
USA to GTM Conversion Process



The workflow has the following activities:

Implement GTM. Run a setup program to implement GTM for US tax processing.

Convert Master Records. Run a second program to populate database tables for customer and supplier addresses, items, product lines, and other master records with GTM data values.

Convert Transaction Records. Run a third program to populate transaction records. All transactions subject to tax are affected, including sales, purchasing, accounts payable, accounts receivable, and service/support management.

Note After each of these activities, review the corresponding reports and audit trails. Mistakes can be pervasive and costly.

Table A.15 lists the eB2.1 and later programs used during the conversion.

Table A.15
eB2.1 and Later Programs Used to Convert US Taxes to GTM

Activity	eB2.1 and Later and Programs
Implementing GTM	USA to GTM–Setup (2.13.22.5)
Converting master records	USA to GTM–Masters (2.13.22.6)
Converting transaction records	USA to GTM–Transactions (2.13.22.7)

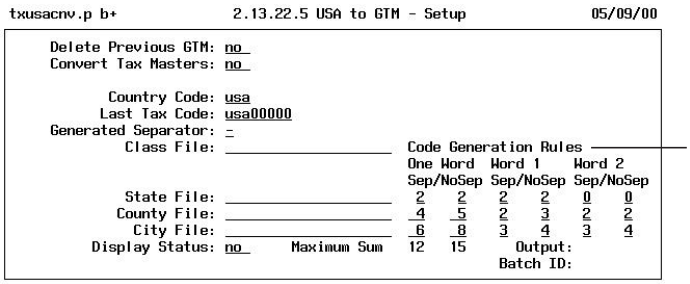
Implementing GTM

An automated setup program creates the codes required to implement GTM, based on how your US taxes are currently defined. Before executing this program, understand the options it provides and the default logic it uses.

Code Generation Rules

GTM codes for tax types, tax zones, and tax environments consist of text strings that uniquely identify the state, county, and city of a tax jurisdiction. Since there are thousands of these codes, setting them up manually would be a tedious process.

Fig. A.11
Code Generation Rules in USA to GTM–Setup (2.13.22.5)



Used to generate codes for Tax Types, Tax Zones, and Tax Environments

Therefore, by default, USA to GTM–Setup creates codes based on a set of rules. These rules systematically select characters from the state code, county name, and city name in the tax master.

To determine if the generated codes are appropriate for your company, run the setup and review the audit trail. If you need a different coding scheme, read the rest of this section and settings for the code generation rules as necessary. Then, rerun the setup with Delete Previous GTM set to Yes and Convert Tax Masters set to Yes.

Table A.16 lists the default generated code formats for US taxes.

Table A.16
Default Generated Code Formats in USA to GTM– Setup (2.13.22.5)

Code	Format	Explanation
Tax zones and tax environments	*SS-CCCC-cccccc*	*SS* is the 2-character state code, *CCCC* is the 4-character county name, and *cccccc* is the 6-character city name. A dash (–) separates each text string.
Tax Types	*SS-CCCC-cccccc-#*	This code format is the same as the previous one, except that tax types have an extra digit (#). The extra digit specifies whether the tax type applies to the state (1), county (2), or city (3).

The rules that determine the characters to select depend on two factors:

- If the US code or name used to generate the text string consists of one word or multiple words (blank spaces separate text).
- Whether separator characters are used

Table A.17 lists the default number of characters for each text string under the different conditions. However, you can change the number of characters and use a different separator or no separator. You can do so as long as the total number of generated characters (including the separator and any ending integers) is 16 or less.

Table A.17
Code Generation Rules in USA to GTM–Setup (2.13.22.5)

	One Word		Multiple Words			
			Word 1		Word 2	
Code	Sep	No Sep	Sep	No Sep	Sep	No Sep
State	2	2	2	2	0	0
County	4	5	2	3	2	2
City	6	8	3	4	3	4
Maximum Total Characters, With Separator						12
Maximum Total Characters, Without Separator						15

Example If the original state code for Arkansas is ARKA, the generated text string is AR regardless of whether separators are used. For the county of Orange, the generated text string is Oran if separators are used and Orang if they are not. For the city of North Hollywood, the generated text string is NorHol if separators are used and NortHoll if they are not.

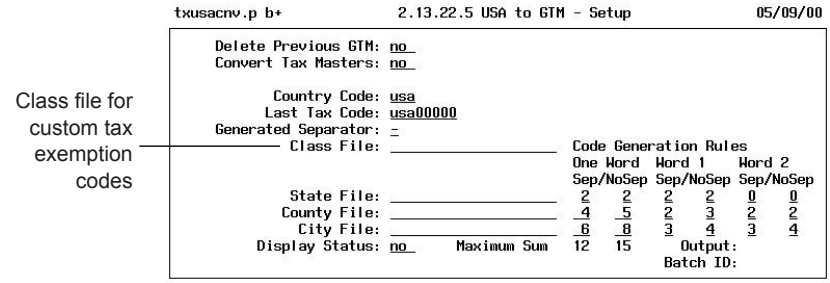
To override duplicate strings such as AR for the states Arkansas and Arizona, see “Defining Custom Codes for States, Counties, and Cities” on page 104.

The setup retains the capitalization from the original US code or name. If the original code or name contains punctuation such as a period, the code generation rules treat it the same as any other non-blank character.

Defining Custom Tax Exemption Codes

By default, the conversion generates corresponding GTM tax classes for your current tax exemption codes. For example, for tax exemption 1, the conversion generates GTM tax class 1. You can override this behavior by creating your own map for the setup program and specifying it in the Class File field. The same class file is also referenced in the programs that convert master data and transaction records.

Fig. A.12
Class File in USA to GTM–Setup (2.13.22.5)



Create a class file if:

- Your company plans to change its tax exemption codes during the GTM conversion.
- You want to convert exemption codes to tax usages instead of tax classes.

The class file is an ASCII file with text strings in the following format:

```
“Current Tax Exemption Code” “GTM Tax Class” “GTM Tax Usage”
```

GTM tax classes can have a maximum of three characters and tax usage codes can have a maximum of eight characters. A null string (“ ” or “”) represents an unused optional value.

The class file can have any name or extension. However, commas instead of blank characters separate code values in .csv files. Place the file in the home directory for the Progress session. A .csv file is a Windows comma-separated value file format that saves recorded values in a spreadsheet.

Example Your current exemption codes are 1 and 2. You want to map these codes to GTM tax classes 01 and 02.

```
“1” “01” “ “
“2” “02” “ “
```

Defining Custom Codes for States, Counties, and Cities

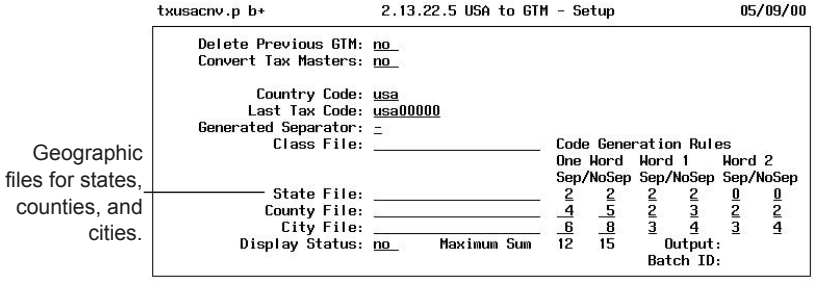
The generated codes for tax zones, tax environments, and tax types consist of text strings that identify the state, county, and city. By default, the code generation rules define the structure of these text strings. However, if you need a different naming convention, you can create geographic files for state codes, county names, and/or city names. See “Updates to Company Addresses” on page 107.

In addition to supporting alternate naming conventions, such files can resolve code generation conflicts. For example, for state codes ARIZ and ARKA, the generated GTM code is AR. A state file is necessary to provide unique values.

Note You only define codes for conditions that the code generation rules do not support.

Create a separate file for each kind of text string and reference the file in USA to GTM–Setup as shown in Figure A.13.

Fig. A.13
State, County, and City Files in USA to GTM–Setup (2.13.22.5)



A geographic file is the same as a class file, except that it uses the following format:

```
“Current Code or Name” “GTM Text String”
```

Example To create unique codes for Arizona and Arkansas, create a state file with these lines:

```
“ARIZ” “AZ”
“ARKA” “AR”
```

Processing Logic

USA to GTM–Setup creates records as described in Table A.18.

Table A.18
New GTM Records

Type of Record	Explanation
Tax zones	Based on code generation rules or a class file, setup builds the tax zone hierarchy for the country and all state/county/city combinations in your current system.
Tax types	Based on code generation rules or a class file, setup generates tax types for all state/county/city combinations in your current system.
Tax environments	Based on code generation rules or a class file, setup generates tax environments for all ship-to tax zones. In the US, the tax environment’s ship-from tax zone is the default country code, and the customer or supplier tax class is blank.
Tax rates	Based on code generation rules or a class file, setup generates tax rates for the tax jurisdictions and percentages in the tax master.

Type of Record	Explanation
Tax classes	By default, setup generates corresponding tax classes for existing US tax exemption codes. However, if you reference a class file, the mappings in the class file determine the actual exemption codes.
Tax usages	By default, the setup does not generate tax usages. However, you can do so in the class file.
Rounding method	The setup enters the rounding method specified in System/Account Control (36.1) as the GTM rounding method in Global Tax Management Control (2.13.24).
Country code	The setup generates a record for the default country code you specify when you run the setup. This country code is the top-level zone in the tax zone hierarchy.
Company addresses and address list types	In GTM, company sites require a corresponding company address record because taxes are calculated using address, not site. The setup verifies that each company site has an address record and creates any missing ones, along with any needed address list type records. The setup also creates a ~taxes address record to provide a default tax address whenever a transaction is missing a company site code.

GTM Control Settings

The setup resets Global Tax Management Control (2.13.24) as described in Table A.19.

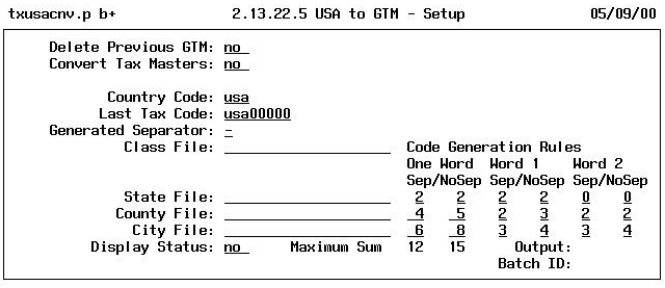
Table A.19
Updates to GTM Control Settings

Field	Explanation
Country Code	As specified during the setup
Tax Method	01
Tax-By-Line	No
Accrue Tax at Receipt	Yes
Discount Tax at Invoice	No
Discount Tax at Payment	No
Last Tax Code	As specified during the setup
Rounding Method	Value specified in System/Account Control (36.1)

USA to GTM–Setup

Based on your implementation decisions, use USA to GTM–Setup (2.13.22.5) to set up GTM for US tax processing.

Fig. A.14
USA to GTM–Setup (2.13.22.5)



Warning As noted previously, the purpose of the conversion is to replicate your existing tax processing setup in GTM. Do not attempt to implement new GTM functionality until after the entire conversion is complete. Do not change settings in the new GTM records or in Global Tax Management Control (2.13.24). If you do, the conversion could fail.

Delete Previous GTM. This option determines whether the setup deletes previously created GTM records from the database. If you select this option, the setup deletes tax zones, tax types, tax environments, transaction tax details, and other GTM records from the database.

- Enter No if you have not yet converted your database to GTM.
- Enter Yes to clean up the database if it contains GTM records from unsuccessful conversion or installation attempts.

Convert Tax Masters. This option determines whether the setup generates the GTM master records from USA tax master records.

- Enter Yes to create corresponding GTM records for the USA tax master: tax classes, tax types, tax zones, tax environments, tax rates, and so on. The audit trail shows the USA tax master records and the new tax zone codes.
- Enter No if you only want to delete previous GTM records and do not want the setup program to generate new GTM records.

Country Code. This country code is the top-level tax zone in the tax zone hierarchy. All other tax zones sum into this one.

If Global Tax Management Control (2.13.24) already has a country code, it displays here. Otherwise, the setup sets the default country code to USA. If you override the value here, the setup assigns it to the control program.

Last Tax Code. Enter a value to update the corresponding field in Global Tax Management Control. In GTM, tax codes identify individual tax rates. Codes are generated sequentially based on the value of last tax code in the GTM control program.

The default Last Tax Code is an 8-character value that consists of the GTM country code and a right-justified integer with placeholder zeros. For example, for country code USA, the default Last Tax Code is USA00000. The system assigns the number USA00001 to the first tax rate record created in GTM and increments this number for subsequent rates.

If you want tax codes to have a different format, enter a different prefix. Codes display alphanumerically in screens and reports. Tax codes that are numeric are left justified and have no placeholder zeros. For example, codes 1 through 30 display in a report column as follows:

```
1
...
19
2
20
...
30
```

Generated Separator. Enter a character to use as a separator in system-generated tax zone, type, and environment codes. Using a separator improves readability of the elements of these codes.

The default separator is the dash (–), but you can enter any character. A sample GTM code that uses the dash separator is CA–SBa–SBa for Santa Barbara, California. If you do not want to use separators, enter blank. However, you cannot use blank as a separator character.

Class File. To provide custom mapping of US tax exemption codes to GTM classes and usage codes, specify an ASCII file with conversion information.

State, County, City File. To override default code generation rules, specify specific values for geographic locations in an ASCII file.

Code Generation Rules. Enter appropriate values for your organization.

Display Status. This setting determines whether the system displays status messages online during the conversion. These messages list database tables and their indexes as they are converted. If you select this option, messages display on the screen and the printed report.

Updates to Company Addresses

In GTM, company sites require a corresponding company address record because taxes are calculated using address, not site. The setup creates any missing company address records for company sites. However, the setup does not populate these new address records with the city, county, state, and country. Supply this information manually in Company Address Maintenance (2.12).

If the setup did not generate codes for these tax jurisdictions, you must also set up tax zone codes to support these new addresses. Do this setup in Tax Zone Maintenance (2.13.3.13). Then, assign the tax zone to the address.

Setup Audit Trail

USA to GTM–Setup prints an audit trail of updated tax master records. For each record, the report shows the state/county/city combination, tax effective date, the tax rates for the effective date, and the taxable status of trailer charges. It also shows the corresponding generated tax zone and its sums-into tax zone.

Figure A.15 shows the audit trail format.

Fig. A.15
Setup Audit Trail

```
Processing: Create GTM from tax masters
                                                    Tax                Sums-Into
State County City      Effective  Tax    Tax    Tax  Trl Tax Zone        Tax Zone
----- ------ --------- --------- ------ ------ ------ --- -------------- --------
FL    ORANGE KISSIMMEE 08/07/97  8.00% 2.00% 6.00% NO  FL-ORAN-KISSIM USA
FL    ORANGE ORLANDO   10/17/92  7.00% 0.00% 9.00% NO  FL-ORAN-ORLAND USA
```

Troubleshooting GTM Setup

After you run USA to GTM–Setup, verify that the GTM setup is correct before you continue with the conversion. The errors listed in Table A.20 can cause issues or unexpected values. Before you proceed to the master conversion, review the audit trail, the GTM reports for the new records, and Global Tax Management Control settings. Correct any errors before continuing.

Subsequent setups do not automatically overwrite the records previous ones created. To set up new records, first delete the old ones. If you rerun the setup, remove the records the earlier setup attempt created by setting Delete Previous GTM to Yes and Convert Tax Masters to Yes.

Note If you must rerun the setup after you have run any of the other conversion programs, restore the database first. Then, rerun the setup and any other conversion programs you ran previously. This action is necessary to propagate changes to master data, transactions, and tax details.

Table A.20
Troubleshooting the GTM Setup

Error	Explanation
Tax system must be USA.	The USA to GTM setup can be run only on a US tax system.
Must delete previous GTM when converting.	When you set Convert Tax Masters to Yes, also set Delete Previous GTM to Yes.
Tax-trailers has changed, cannot convert prior to this date.	The setup can convert only the current tax environment, not previous variations. If the taxable status of trailer charge codes changed in the time span included in the conversion, the setup creates tax environments only for current conditions.

Warning Do not correct records programmatically. This approach often causes additional problems.

How the Setup Changes GTM Records

The following is a technical description of how the setup updates the database.

The menu-level program for USA to GTM–Setup is `txusacnv.p`. This program calls subprograms (primarily `txusatax.p`) that set the GTM tax values in the individual database records. Table A.21 lists the affected database tables and summarizes the changes.

Table A.21
Changes to GTM Records

Tables	Summary of Changes
Address Master (ad_mstr)	`txusatax.p` creates one ~taxes record for the database. It also scans si_mstr and creates an address record for any company site that does not already have one.
Generalized Code Master (code_mstr)	For the nine US exemption codes in tax_mstr, `txusatax.p` creates corresponding tax classes. If a class file is referenced, it creates the specified tax classes. For each of the three rates that tax_mstr stores for US tax jurisdictions, `txusatax.p` creates a tax type. For non-taxable transactions, it also creates a default NON-TAX tax type. Finally, for each ship-to tax zone, `txusatax.p` creates a tax environment and assigns it the tax types associated with the tax zone.
Country Master (ctry_mstr)	`txusatax.p` creates a record for the default country code specified in the selection data.
Address List Detail (ls_mstr)	`txusatax.p` creates ls_mstr record for each new ad_mstr record, if any.
Tax Master (tx2_mstr)	For each of the three rates that tax_mstr stores for US tax jurisdictions, `txusatax.p` creates a tax rate. It also runs txtx2_nt.i to create a non-taxable tax rate and txtxmeth.i to create tax method 01.
Tax Control (txc_ctrl)	`txusatax.p` sets txc_ctry_code and txc_tax_code from the selection data. It sets txc_method to 01, txc_by_line, txc_inv_disc, and txc_pmt_disc to No, and txc_rcpt_tax_point to Yes.
Tax Environment Master (txe_mstr)	`txusatax.p` creates tax environment zone detail records for every tax environment code it generates for the code_mstr.

Tables	Summary of Changes
Tax Environment Detail (txed_det)	`txusatax.p` creates tax environment tax type detail records for every tax environment code it generates for the code_mstr.
Tax Zone Master (txz_mstr)	`txusatax.p` creates a top level sums-into tax zone for the new ctry_mstr record. For each state/county/city combination in tax_mstr, it creates a ship-to tax zone.

Converting Master Records

Once you finish the GTM setup, the next activity is to update tax settings in the following master records:

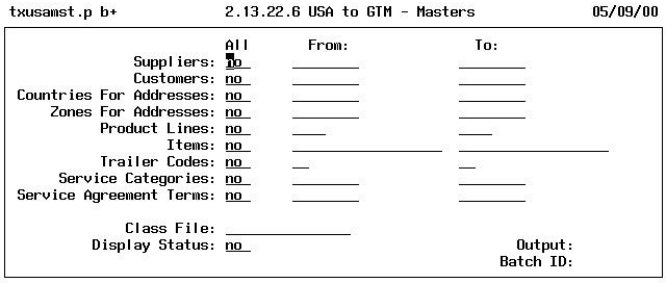
- Suppliers
- Customers
- Product Lines
- Items
- Trailer Codes
- Service Categories
- Service Agreement Terms

GTM has additional fields and could require new values for existing fields.

USA to GTM–Masters

To convert master records, run USA to GTM–Masters (2.13.22.6).

Fig. A.16
USA to GTM–Masters (2.13.22.6)



In addition to updating the tax settings in the master records, this program assigns tax zone codes to supplier, customer, and company address records. For verification of changes, the program generates an audit trail.

Important Before you run USA to GTM–Masters, do the following:

- Run USA to GTM–Setup (2.13.22.5).
- To avoid record contention conflicts with other users, shut down the database. Restart it when no other users are on the system.

For each type of record, you can convert all records, a range of records, or individual records. This program converts records in the same order as the options on the screen. For separate audit trail reports, run the report separately for each type of record.

If you created a class file during the setup step, specify its name in the Class File field. Display Status, Output, and Batch ID are the same as in the setup program.

See “Defining Custom Tax Exemption Codes” on page 103.

Master Conversion Audit Trail

The master conversion prints a report of changed records. The format varies depending on the records included in the conversion. For each group of converted records, the report shows the record number and name and the before and after tax information. The tax information includes the country code, tax zone code, taxable status, whether tax is included in item amounts, and tax class.

Groups of converted records print in the same order as the screen selection criteria with a page break separating each group. Warning and error messages identify potential conversion issues.

Figure A.17 and Figure A.18 show representative audit trail formats.

Fig. A.17
Customer Audit Trail

```
Processing: Customers

Address  Name                                 Taxable TxC TaxUsage
-------- ---------------------------          ------- --- --------
32174893 Consolidated Industries Inc. Before  No       1
                                      After   No       1
32174895 Asheville Manufacturing      Before  Yes
                                      After   Yes
32174897 Hartford Electronics         Before  Yes
                                      After   Yes
```

Fig. A.18
Countries for Addresses Audit Trail

```
Processing: Countries For Addresses

Address  Name                                 Ctry Country
-------- ---------------------------          ---- -------------
32174893 Consolidated Industries Inc. Before       United States
                                      After   USA  United States
32174895 Asheville Manufacturing      Before       United States
                                      After   USA  United States
32174897 Hartford Electronics         Before       United States
                                      After   USA  United States
```

Troubleshooting the Master Conversion

The error messages in the audit trail identify conditions to analyze and address before converting transactions. Table A.22 lists some common errors along with explanations. Before you make corrections, restore the database from backup.

Warning Do not proceed to the transaction conversion until the master conversion audit trail is free of errors.

Table A.22
Troubleshooting the Master Conversion

Error	Explanation
Tax class cannot be converted.	Class file does not contain a tax exemption that matches the one in the master record.
Blank tax class not allowed.	Tax exemption is blank in the class file.

Error	Explanation
Tax class cannot exceed 3 characters (*xxx*).	Tax exemption in the class file is longer than three characters. The message shows the first three characters.
Tax class does not exist (*x*).	Tax exemption in the class file not in the US tax master.
Tax class is not unique (*x*).	Tax exemption occurs in multiple places in the class file.
Tax class does not exist (*xxx*).	GTM tax class in the class file does not exist in the GTM tax class master.
x, *xxx*, and *xxxxxxxx* are placeholders for the actual codes displayed in the error message.	

How the Conversion Changes Master Records

The following is a detailed description of how the master conversion updates the database.

The menu-level program for USA to GTM–Masters, `txusamst.p`, sets the GTM tax values in the individual tables.

When setting the GTM tax class value, the programs can set the existing tax exemption code or retrieve an alternate from a class file.

See “Defining Custom Tax Exemption Codes” on page 103.

Table A.23 lists the affected database tables and summarizes the changes.

Table A.23
Changes to Master Records

Tables	Summary of Changes
Address Master (ad_mstr)	In supplier records, `txusavd.p` sets ad_taxable from vd_taxable. It also sets ad_taxc and ad_tax_usage to blank. In customer records, `txusacm.p` sets ad_taxable from cm_taxable and ad_tax_in from cm_tax_in. It also sets cm_taxc, ad_taxc, cm_taxc, and ad_tax_usage from cm_taxc or from the AR class file, if any. In all address records, `txusact.p` sets ad_ctry from ad_country and the other way around. `txusazn.p` calls `txtxzget.p` to set ad_tax_zone.
Service Category Master (fsc_mstr)	`txusafsc.p` sets fsc_taxc from fsc_taxc or from the AR class file if any.
Product Line Master (pl_mstr)	`txusapl.p` sets pl_taxc from pl_taxc or from the class file, if any.
Item Master (pt_mstr)	`txusapt.p` sets pt_taxc from pt_taxc or from the class file, if any.
Service Agreement Terms and Conditions Master (sv_mstr)	`txusasv.p` sets sv_taxc and sv_tax_usage from sv_taxc or from the AR class file, if any.
Trailer Master (trl_mstr)	`txusatrl.p` sets trl_taxc from trl_taxc or from the class file, if any.

Converting Transaction Records

Once you finish converting master records, you can convert the following transaction records:

- Purchase orders and receipts
- Accounts payable vouchers and payments
- Service contracts, calls, and return material authorizations
- Sales quotes and orders
- Accounts receivable memos, invoices, and payments

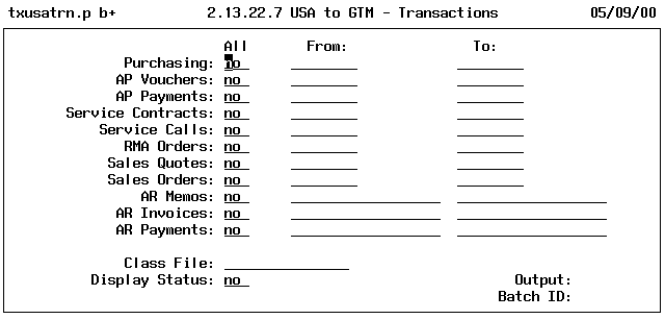
In GTM, every transaction subject to tax has a transaction tax detail record. This record stores the information used to calculate tax.

Note If you already have GTM transaction records in the database when you perform transaction conversion, the conversion process updates them using the current default tax values. Values replaced in these records include tax environment, class, usage, and so on.

USA to GTM–Transactions

To convert existing transaction records, run USA to GTM–Transactions (2.13.22.7).

Fig. A.19
USA to GTM–Transactions (2.13.22.7)



In addition to updating transactions, this program generates an audit trail for verification of changes.

Important Before you run USA to GTM–Transactions, do the following:

- Run USA to GTM–Setup (2.13.22.5) and USA to GTM–Masters (2.13.22.6).
- Avoid record-contention conflicts with other users. Shut down the database and restart it when no other users are on the system.

You can convert all records, a range of records, or individual records. The program converts records in the same order they display on the screen. For separate audit trail reports, run the report separately for each type of record.

Note In some cases, the record sequence is important. Purchasing transactions are converted before accounts payable vouchers and vouchers before payments. Accounts receivable memos and invoices are converted before payments.

If you created a class file during the setup step, specify its name in the Class File field. Display Status, Output, and Batch ID are the same as in the setup program.

See “Defining Custom Tax Exemption Codes” on page 103.

Transaction Audit Trail

The transaction conversion prints a report of changed records. The format varies depending on the records included in the conversion. For each group of converted records, the report shows the transaction number and name and the before and after tax information for each line item. The tax information includes taxable status, tax environment, and tax class. Groups of converted records print in the same order as the screen selection criteria with a page break separating each group.

Warning and error messages identify potential conversion issues. Messages that appear at the end of a transaction apply to the entire transaction; messages that appear between the Before and After line apply only to that line.

Figure A.20 and Figure A.21 show representative audit trail formats.

Fig. A.20
Purchasing Audit Trail

```
Processing: Purchasing

Order    Receiver  Ln         Tax TxC TaxUsage Tax Env  prh_tax_at
-------- -------- ---        --- --- -------- -------- ----------
01104533             Before  No  1
                     After   No  1            NJ-TRENT
                   1 Before  Yes B
                     After   Yes B            NJ-TRENT
         RC1290    1 Before      B                    B
                     After       B            NJ-TRENT B
```

Fig. A.21
Service Calls Audit Trail

```
Processing: Service Calls

Call ID  Call/SR  Line Record              Tax TxC TaxUsage Tax Env
-------- -------- ---- ----------          --- --- -------- ----------------
CA127                  Call       Before    No  1
                                  After     No  1   1-P-MFG  NJ-TRENT
         CA127       1 Item       Before    No  1
                                  After     No  1   1-P-MFG  NJ-TRENT
         CA127       1 Billing    Before    No  1
                                  After     No  1   1-P-MFG  NJ-TRENT
```

Troubleshooting Transaction Conversion

The error messages in the audit trail identify conditions to analyze and address before resuming live GTM processing. Table A.24 lists a common error along with an explanation. Before you make corrections, restore the database from backup.

Table A.24
Troubleshooting the Transaction Conversion

Error	Explanation
Detail tax environment must match header.	In accounts payable vouchers and accounts receivable debit/credit memos, the tax environment are the same in the header and detail lines.

Warning Do not resume live processing until the transaction conversion audit trail is free of errors. Do not correct transaction records programmatically. This approach often causes additional problems.

In addition to examining the audit trail, review the Tax Detail by Transaction Report (2.13.15.3). This report shows the tax environments, tax types, and tax amounts for the converted records. Verify that the tax calculations are as expected.

Converted transactions can have minor differences in before/after tax amounts. The differences occur because GTM uses a different calculation algorithm or rounding method than your source version. To synchronize the general ledger with converted transactions, record adjusting entries.

To eliminate ambiguity, the audit trail shows before and after values for purchasing, accounts payable, and accounts receivable records by their Progress database field name, not their screen label. For example, the audit trail displays voucher line types in the vod_type column.

As you interpret the audit trail for the transaction conversion, it can be helpful to reference Table A.25 through Table A.29. These tables summarize the nature of before/after tax values.

Note In these tables, quotation marks indicate a value that cannot be translated.

Table A.25
USA to GTM, Purchasing Transactions

Status	Tax System	pod_taxable	prh_tax_at
Taxable	USA	Yes	"y"
	GTM	Yes	"Yes"
Non-taxable	USA	No	"n" or blank[1]
	GTM	No	Blank

1. An item is non-taxable if pod_taxable is n (No). If the transaction is non-taxable, the tax exemption code is optional.

Table A.26
USA to GTM, AP Voucher Receiver Lines

Status	Amt	Tax System	Tax	TxC	vod_type	vod_tax	vod_tax_at
Taxable	Item	USA	No	Blank	"R"	Blank	"y"
		GTM	Yes	Tax class	"R"	Blank	"Yes"
	Tax	USA	No	Blank	Blank	"y"	Blank
		GTM	No	Blank	"T"	"t"	"No"
Non-taxable	Item	USA	No	Exemption code or blank[1]	"R"	Blank	Blank
		GTM	No	Tax class	"R"	Blank	Blank
	Tax[2]	USA	No	Blank	Blank	"y"	Blank
		GTM	–	–	–	–	–

1. An item is non-taxable if its taxable status is No. If the transaction is non-taxable, the tax exemption code is optional.
2. The conversion deletes US tax lines resulting from non-taxable amounts.

Table A.27
USA to GTM, AP Voucher Memo Lines

Status	Amt	Tax System	Tax	TxC	vod_type	vod_tax	vod_tax_at
Taxable	Item	USA	No	Blank	Blank	Blank	Y
		GTM	Yes	Tax class	Blank	Blank	"Yes"
	Tax	USA	No	Blank	Blank	y	Blank
		GTM	No	Blank	"T"	"t"	"No"
Non-taxable	Item	USA	No	Blank	Blank	Blank	n or blank[1]
		GTM	No	Tax class	Blank	Blank	"No"
	Tax[2]	USA	No	Blank	Blank	y	Blank
		GTM	–	–	–	–	–

1. An item is non-taxable if its taxable status is n (No). If the transaction is non-taxable, the tax exemption code is optional.
2. The conversion deletes USA tax lines resulting from non-taxable amounts.

Table A.28
USA to GTM, AR Invoices

Status	Amt	Tax System	ard_tax	ard_tax_at	ard_taxc
Taxable	Item	USA	Blank	Blank	Blank
		GTM	Blank	Tax class	Tax class
	Tax	USA	Blank	Blank	Blank
		GTM	“t”	“No”	Blank
Non-taxable	Item	USA	Blank	Blank	Blank
		GTM	Blank	Tax class	Tax class
	Tax[1]	USA	–	–	–
		GTM	–	–	–

1. In the US tax system, no tax records are created for non-taxable amounts, so the conversion creates no records for GTM.

Table A.29
USA to GTM, AR DR/CR Memos

Status	Amt	Tax System	ard_tax	ard_tax_at	ard_taxc
Taxable	Item	USA	Blank	“y”	Blank
		GTM	Blank	“Yes”	Tax class
	Tax	USA	“y”	Blank	Blank
		GTM	“t”	“No”	Blank
Non-taxable	Item	USA	Blank	“n”	Blank
		GTM	Blank	“No”	Tax class
	Tax[1]	USA	–	–	–
		GTM	–	–	–

1. In the US tax system, no tax records are created for non-taxable amounts, so the conversion creates no records for GTM.

How the Conversion Changes Transaction Records

The following is a detailed description of how the transaction conversion updates the database.

The menu-level program for USA to GTM–Transactions is `txusatrn.p`. This program calls subprograms that set the GTM tax values in the individual database records. For all transactions, the conversion also generates corresponding tax detail records in the Tax Detail (tx2d_det) table.

When setting the GTM tax class value, these programs can set the existing tax exemption code value or retrieve an alternate value from a class file.

See “Defining Custom Tax Exemption Codes” on page 103.

Table A.30 lists the affected database tables and summarizes the changes.

Table A.30
Changes to Transaction Records

Tables	Summary of Changes
Accounts Receivable Detail (ard_det)	For debit/credit memos, `txusaarm.p` sets ard_tax and ard_tax_at.
Accounts Receivable Master (ar_mstr)	For debit/credit memos, `txusaarm.p` sets ar_tax_env using `txtxeget.p`.

Tables	Summary of Changes
Service/Support Call Master (ca_mstr)	`txusaca.p` sets ca_taxc and ca_tax_usage from ca_taxc or from the class file, if any. It also sets ca_tax_env using `txtxeget.p`.
Invoice History Detail (idh_hist)	`txusaari.p` sets idh_taxc and idh_tax_usage from idh_taxc or from the class file, if any. It also sets idh_tax_env using `txtxeget.p`.
Invoice History Master (ih_hist)	`txusaari.p` sets ih_taxc and ih_tax_usage from ih_taxc or from the class file, if any. It also sets ih_tax_env using `txtxeget.p`.
Service/Support Call Item Detail (itm_det)	If itm_prefix is CA and itm_type is any value except INV, `txusaca.p` sets itm_taxc and itm_tax_usage from itm_taxc or from the class file, if any. It also sets itm_tax_env using `txtxeget.p`.
Purchase Order Detail (pod_det)	If pod_taxable is Yes, `txusapo.p` sets pod_taxc to blank and sets pod_tax_env using `txtxeget.p`. If pod_taxable is No, `txusapo.p` sets pod_tax_env from po_tax_env.
Purchase Order Master (po_mstr)	`txusapo.p` sets po_tax_pct[1], po_tax_pct[2], and po_tax_pct[3] to 0. It sets po_tax_usage from ad_tax_usage and po_taxc from ad_taxc. It also sets po_tax_env using `txtxeget.p`.
Purchase Order Receipt History (prh_hist)	`txusapo.p` sets prh_taxc to blank when prh_tax_at is “y.” It also sets prh_tax_env from pod_tax_env. If pod_taxable is Yes, `txusapo.p` sets prh_tax_at to “Yes”; otherwise, it is set to blank.
Sales Quotation Detail (qod_det)	`txusaqo.p` sets qod_taxc and qod_tax_usage from qod_taxc or from the class file, if any. It also sets qod_tax_env using `txtxeget.p`.
Sales Quotation Master (qo_mstr)	`txusaqo.p` sets qo_tax_pct[1], qo_tax_pct[2], and qo_tax_pct[3] to 0. `txusaqo.p` also sets qo_taxc and qo_tax_usage from qo_taxc or from the class file, if any. It also sets qo_tax_env using `txtxeget.p`.
Return Material Authorization Master (rma_mstr)	`txusarma.p` sets rma_taxc from rma_taxc or from the class file, if any.
Service Contract Detail (sad_det)	`txusasc.p` sets sad_taxc and sad_tax_usage from sad_taxc or from the class file, if any. `txusasc.p` sets sad_tax_env from sa_site and sa_taxc using `txtxeget.p`.
Service Contract Master (sa_mstr)	`txusasc.p` sets sa_tax_pct[1], sa_tax_pct[2], and sa_tax_pct[3] to 0. It sets sa_taxc and sa_tax_usage from sa_taxc or from the class file, if any. It also sets sa_tax_env using `txtxeget.p`.
Service/Support Billing Detail (sfb_det)	`txusaca.p` sets sfb_taxc and sfb_tax_usage from sfb_taxc or from the class file, if any. `txusaca.p` sets sfb_tax_env using `txtxeget.p`.
Sales Order Detail (sod_det)	`txusaso.p` sets sod_taxc and sod_tax_usage from sod_taxc or from the class file if any. `txusaso.p` sets sod_tax_env using `txtxeget.p`.
Sales Order Master (so_mstr)	`txusaso.p` sets so_tax_pct[1], so_tax_pct[2], and so_tax_pct[3] to 0. `txusaso.p` sets so_taxc and so_tax_usage from so_taxc or from the class file, if any. `txusaso.p` sets so_tax_env using `txtxeget.p`.

Tables	Summary of Changes
Tax Detail (tx2d_det)	`txusapo.p` creates tax details for purchase orders (GTM transaction type 20), receivers (21), reconciliations (23), and returns (25). `txusaapv.p` creates tax details for vouchers (22) and recurring vouchers (32). `txusaapp.p` creates tax details for accounts payable tax on discount at payment (29). `txusaqo.p` creates tax details for sales quotes (10). `txusaso.p` creates tax details for invoiced service calls (38), return material authorizations (36), sales orders (11), and pending invoices (13). `txusaarm.p` creates tax details for debit/credit memos (18). `txusaari.p` creates tax details for invoices (16). `txusaarp.p` creates tax details for accounts receivable tax on discount at payment (19). `txusasc.p` creates tax details for service quotes (33) and service contracts (34).
Voucher Detail (vod_det)	`txusaapv.p` sets vod_taxable, vod_taxc, vod_tax_usage, vod_tax, vod_tax_at, vod_type, and vod_tax_env. If vod_type is “r” and vod_tax_at is “Y”, vod_taxc is set to blank.
Voucher Master (vo_mstr)	For vouchers, `txusaapv.p` sets vo_tax_pct[1], vo_tax_pct[2], and vo_tax_pct[3] to 0. It also sets vo_taxable, vo_taxc, vo_tax_usage, and vo_tax_env.

Converting to GTM From No Taxes

If you are not using any tax method in your source database, follow the steps in "Converting US Taxes to GTM" on page 101 to convert master records and transactions details. The modifications to the process documented in that section are as follows:

USA to GTM Setup

Under "Implementing GTM" on page 101, when you run USA-to-GTM– Setup (2.13.22.5), use the following settings:

Table A.31
USA-to-GTM Setup

Field	Value
Delete Previous GTM	Yes
Convert Tax Masters	Yes
Country Code	Non-blank
Last Tax Code	USA00000
Display Status	Yes

Accept default values for all other fields.

Enter any new country codes entered in US-to-GTM Setup in Country Code Maintenance (2.14.1). In addition, create a country code called All with a name of All. The Name field in Country Code Maintenance must match the Country field in the Address Master (ad_mstr) table.

USA to GTM Masters

Under "Converting Master Records" on page 109, run the following programs:

- Tax Zone Maintenance (2.13.3.13). Create a Tax Zone All with a country code of All. In addition, create tax zones for each country code. Set the option Sums-Into Tax Zone to All.
- Tax Environment (2.13.5.1). Create a non-taxable environment with the following setup:

Table A.32
Non-taxable Environment Setup

	Field	Value
	Tax Environment	non-tax
First record:	Ship-From Zone	All
	Ship-To Zone	blank
Second record:	Ship-From Zone	blank
	Ship-To Zone	All

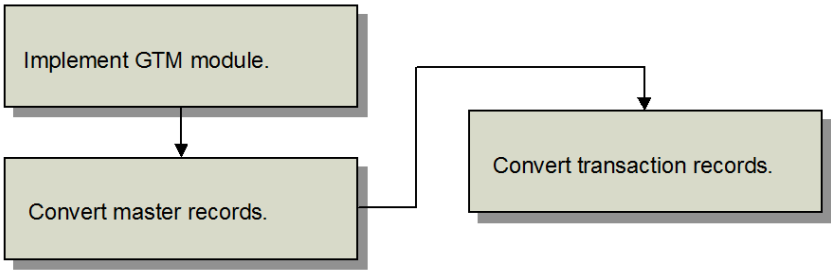
- Global Tax Management Control (2.13.24). Enter the default tax zone and tax environment. Set Tax Zone to the zone created for your country and set Tax Environment to `non-tax`.

Continue on with "USA to GTM–Masters" on page 109 and "USA to GTM–Transactions" on page 112.

Converting Canadian Taxes to GTM

This section describes how to convert to GTM from MFG/PRO's Canadian tax system. The Canadian to GTM conversion process translates Canadian tax data to GTM equivalents and updates existing eB2.1 and later records. Figure A.22 summarizes the conversion workflow.

Fig. A.22
Canadian to GTM Conversion Process



This workflow revolves around three sets of activities:

Implement GTM Module. You run a setup program to implement GTM for Canadian tax processing.

Convert Master Records. You run a second program to populate database tables for customer and supplier addresses, items, product lines, and other master records with GTM data values.

Convert Transaction Records. You run a third program to populate transaction records. All transactions subject to tax are affected, including sales, purchasing, accounts payable, accounts receivable, and service/support management.

Note After each of these activities, it is crucial to review the corresponding reports and audit trails. Mistakes can be pervasive and costly.

Table A.33 lists the eB2.1 and later programs used during the conversion.

Table A.33
eB2.1 and Later Programs Used to Convert Canadian Taxes to GTM

Activity	eB2.1 and Later Programs
Implementing GTM	CAN to GTM–Setup (2.13.22.9)
Converting master records	CAN to GTM–Masters (2.13.22.10)
Converting transaction records	CAN to GTM–Transactions (2.13.22.11)

Implementing GTM

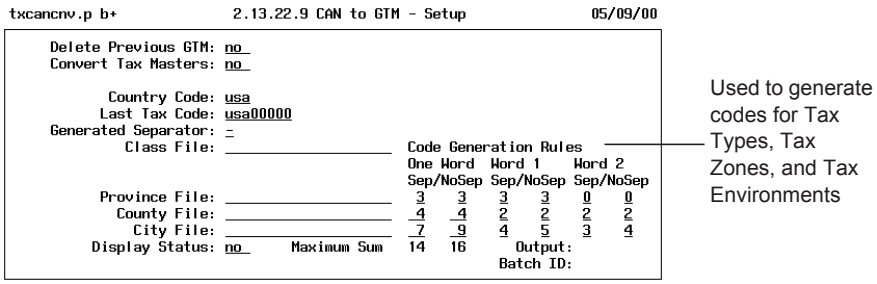
An automated setup program uses your Canadian tax definitions to create most of the codes required to implement GTM. Before executing this program, understand the options it provides and the default logic it uses.

Code Generation Rules

GTM codes for tax types, tax zones, and tax environments consist of text strings that uniquely identify the province, county, and city of a tax jurisdiction. In Canada, there are thousands of these codes; setting them up manually would be a tedious process.

Therefore, by default, CAN to GTM–Setup generates codes based on its code generation rules (Figure A.23). These rules systematically select characters from the province code, county name, and city name in the tax master.

Fig. A.23
Code Generation Rules in CAN to GTM-Setup (2.13.22.9)



To determine if generated codes are suitable for your company, run the setup and review the audit trail. If you need a different coding scheme, read the rest of this section and change the settings for the code generation rules as necessary. Then, rerun the setup with Delete Previous GTM set to Yes and Convert Tax Masters set to Yes.

For Canadian taxes, the default generated code format for tax zones, tax types, and tax environments is:

```
PPP-CCCC-ccccccc
```

PPP is the 3-character province code, *CCCC* the 4-character county or district name, and *ccccccc* the 7-character city name. A dash (–) separates each text string.

The rules used to determine the characters to select depends on:

- If the Canadian code or name used to generate the text string consists of one word or multiple words (blank spaces separate text strings)
- If separator characters are used

Table A.34 lists the default number of characters for each text string under the different conditions. However, you can change the number of characters and use a different separator or no separator. You can do so as long as the total number of generated characters—including the separator and any ending integers—is 16 or less.

Table A.34
Code Generation Rules in CAN to GTM–Setup (2.13.22.9)

	One Word		Multiple Words			
			Word 1		Word 2	
Code	**Sep**	**No Sep**	**Sep**	**No Sep**	**Sep**	**No Sep**
Province	3	3	3	3	0	0
County	4	4	2	2	2	2
City	7	9	4	5	3	4
Maximum Total Characters, With Separator						14
Maximum Total Characters, Without Separator						16

Example If the original province code for Alberta is ALBA, the generated text string is ALB regardless of whether separators are used. For the city of Southampton, the generated text string is Southam if separators are used and Southampt if they are not. For the city of Thunder Bay, the generated text string is ThunBay if separators are used and ThundBay if they are not.

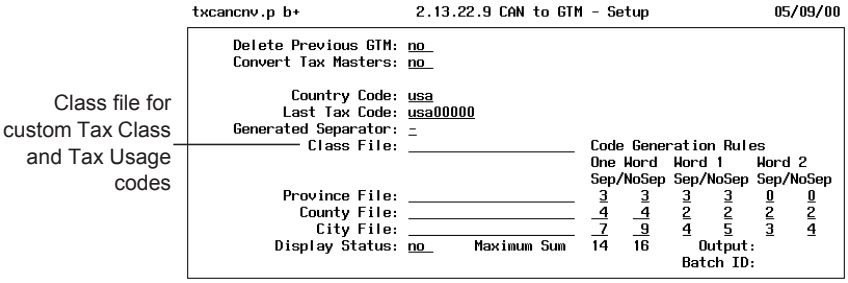
Setup retains the capitalization from the original Canadian code or name. If the original code or name contains punctuation such as a period, the code generation rules treat it the same as any other non-blank character.

Defining Custom Tax Class and Tax Usage Codes

By default, the setup generates tax classes that directly correspond to your Canadian GST master records. To distinguish the two sets of codes, it appends the letter P to the GST + PST code. For example, if your current GST classes are 0, 1, and 2, the generated GTM classes are 0, 1, and 2 (for GST only) and 0P, 1P, and 2P (for GST + PST). The system does not automatically generate tax usages.

You can override this behavior by creating your own map for the setup program and specifying it in the Class File field. The same class file is also referenced in the programs that convert master data and transaction records.

Fig. A.24
Class File in CAN to GTM–Setup (2.13.22.9)



Create a class file if:

- Your company plans to change its tax class codes during the GTM conversion. In this situation, you define custom codes for GST only and GST + PST.
- Within a tax class, a company can be taxed based on its nature of operation or the way it intends to use an item. Tax usage codes identify these conditions in GTM.

The class file is an ASCII file with text strings in the following format:

```
"Current GST Class" "GST Class" "GST Usage" "Class for GST + PST" "Usage for GST + PST"
```

GTM tax classes can have a maximum of three characters, and tax usage codes eight characters. A null string (" " or "") represents an unused value.

The class file can have any name or extension. However, commas instead of blank characters separate code values in .csv files. Place the file in the home directory for the Progress session. A .csv file is a Windows comma-separated value file format that saves recorded values in a spreadsheet.

Example Your current GST codes are 0, 1, and 2, and you want to change them to A, B, and C, plus add a G to indicate GST only.

```
"0" "AG" "" "A" ""
"1" "BG" "" "B" ""
"2" "CG" "" "C" ""
```

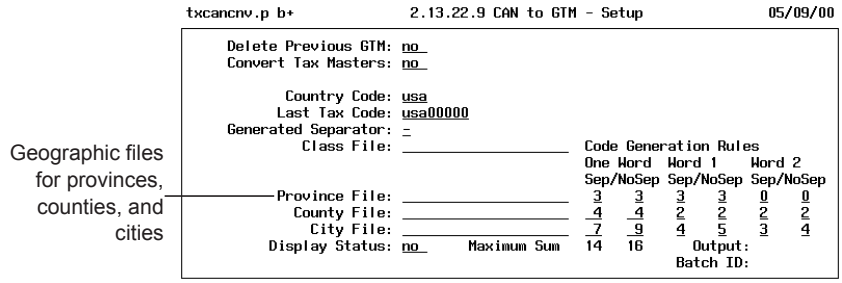
Defining Custom Codes for Provinces, Counties, and Cities

The generated codes for tax zones, tax environments, and tax types consist of text strings that identify the province, county, and city. By default, the code generation rules define the structure of these text strings. However, if you need a different naming convention, you can create geographic files for province codes, county names, and/or city names.

See "Code Generation Rules" on page 119.

Create a separate file for each kind of text string and reference the file in CAN to GTM–Setup as shown in Figure A.25.

Fig. A.25
Province, County, and City Files in CAN to GTM–Setup (2.13.22.9)



Note You only define codes for conditions that the code generation rules do not support.

A geographic file is the same as a class file, except that it uses the format:

```
"Current Code or Name" "GTM Text String"
```

Example To map province codes for the provinces of Manitoba and Quebec, create a province file with these lines:

```
"MANI" "MB"
"QUE" "QB"
```

Processing Logic

This program creates eB2.1 and later records as described in Table A.35.

Table A.35
New GTM Records

Type of Record	Explanation
Tax zones	Based on code generation rules or a class file, setup builds the tax zone hierarchy for the country and all province/county/city combinations used in your current system.
Tax types	Based on code generation rules or a class file, setup generates tax types for the province/county/city combinations used in your current system.
Tax environments	Based on code generation rules or a class file, setup generates tax environments for all ship-to tax zones.
Tax rates	Based on code generation rules or a class file, setup generates tax rates for the tax jurisdictions and percentages used in your current system.

Type of Record	Explanation
Tax classes	By default, the setup generates tax classes based on your Canadian GST master records, in which GST and PST are represented as separate codes. However, the GTM setup generates tax classes for GST only and GST + PST. It sets up PST only as GST + PST with a GST tax rate of 0%. It sets up non-taxable as GST only with GST and PST tax rates of 0%. To distinguish the generated codes, the setup appends the letter P to the GST + PST code. For example, if your current GST classes are 0, 1, and 2, the generated GTM classes are 0, 1, and 2 (for GST only) and 0P, 1P, and 2P (for GST + PST). The system does not automatically generate tax usages. You can choose to bypass the default setup behavior by defining a class file. For more information, see “Defining Custom Codes for Provinces, Counties, and Cities” on page 122.
Tax usages	By default, setup does not generate tax usages. However, you can generate tax usages with a class file.
Country code	The setup generates a record for the default country code you specify when you run the setup. This country code is the top-level zone in the tax zone hierarchy.
Company addresses and address list types	In GTM, company sites require a corresponding company address record because address, not site, calculates taxes. The setup verifies that each company site has an address record and creates any missing ones, along with any needed address list type records. The setup also creates a ~taxes address record to provide a default tax address whenever a transaction is missing a company site code.

GTM Control Settings

Setup defines Global Tax Management Control (2.13.24) as described in Table A.36.

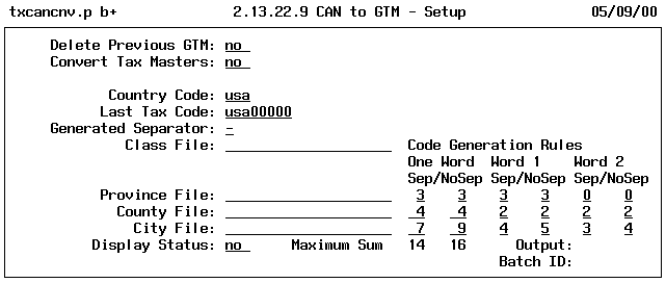
Table A.36
Updates to GTM Control Settings

Field	Explanation
Country Code	As specified during the setup
Tax Method	01
Tax-By-Line	No
Accrue Tax at Receipt	Yes
Discount Tax at Invoice	No
Discount Tax at Payment	No
Last Tax Code	As specified during the setup
Rounding Method	Value specified in System/Account Control (36.1)

CAN to GTM–Setup

Based on your implementation decisions, use CAN to GTM–Setup (2.13.22.9) to set up GTM for Canadian tax processing.

Fig. A.26
CAN to GTM–Setup (2.13.22.9)



Warning As noted previously, the purpose of the conversion is to replicate your existing tax processing setup in GTM. Do not attempt to implement new GTM functionality until after the entire conversion is complete. Do not change settings in the new GTM records or in Global Tax Management Control (2.13.24). If you do, the master and/or transaction conversions could fail.

Delete Previous GTM. This option determines whether the setup deletes previously created GTM records from the database. If you select this option, the setup deletes tax zones, tax types, tax environments, transaction tax details, and other GTM records from the database.

- Enter No if you have not yet converted your database to GTM.
- Enter Yes to clean up the database if it contains GTM records from unsuccessful conversion or installation attempts.

Convert Tax Masters. This option determines whether the setup generates the GTM master records from the master tables for Canadian GST and PST.

- Enter Yes to create GTM records corresponding to Canadian tax masters: tax classes, tax types, tax zones, tax environments, tax rates, and so on. The audit trail shows the new GTM records.
- Enter No if you only want to delete previous GTM records and do not want the setup program to generate new GTM records.

Country Code. This country code is the top-level tax zone in the tax zone hierarchy. All other tax zones sum into this one.

If Global Tax Management Control already has a country code, it displays here. Otherwise, the setup sets the default country code to CAN. If you override the value here, the setup assigns it to the control program.

Last Tax Code. Enter a value to update the corresponding field in Global Tax Management Control. In GTM, tax codes identify individual tax rates. Codes are generated sequentially based on the value of Last Tax Code in the GTM control program.

The default Last Tax Code is an 8-character value that consists of the GTM country code and a right-justified integer with placeholder zeros. For example, for country code CAN, the default Last Tax Code is CAN00000. The system assigns the number CAN00001 to the first tax rate record created in GTM and increments this number for subsequent rates.

If you want tax codes to have a different format, enter a different prefix. Codes display alphanumerically in screens and reports. Tax codes that are numeric are left justified and have no placeholder zeros. For example, codes 1 through 30 display in a report column as follows:

```
1
...
19
2
20
...
30
```

Generated Separator. Enter a character to use as a separator in system-generated tax zones, types, and environments. Using a separator can improve the readability of the component elements of these codes.

The default separator is the dash (–), but you can enter any character. A sample GTM code that uses the dash separator is BC–Van for Vancouver, British Columbia. If you do not want to use separators in codes, enter blank. However, you cannot use blank as a separator character.

Class File. To provide custom mapping of Canadian tax classes to GTM classes and usage codes, specify an ASCII file with conversion information.

Province, County, City File. To override default code generation rules, specify specific values for geographic locations in an ASCII file.

Code Generation Rules. Enter appropriate values for your organization.

Display Status. This setting determines whether the system displays status messages online during the conversion. These messages list database tables and their indexes as they are converted. If you select this option, messages display on the screen and the printed report.

Updates to Company Addresses

In GTM, company sites require a corresponding company address record because taxes are calculated using address, not site. The setup creates any missing company address records for company sites. However, the setup does not populate these new address records with the city, county, province, and country. Supply this information manually in Company Address Maintenance (2.12).

If the setup did not generate codes for these tax jurisdictions, you must set up tax zone codes to support these new addresses. Do the setup in Tax Zone Maintenance (2.13.3.13). Then, assign the tax zone to the address.

Set Up Audit Trail

CAN to GTM–Setup prints a report of updated tax master records for GST and PST rates. For GST rates, the report shows the GST class, description, starting and ending effective dates, the GST percent, and the general ledger tax accounts for accounts payable and accounts receivable. For PST, the report shows the province/county/city, tax effective date, tax rate for the effective date, whether PST is based on GST, and the generated GTM tax zone and sums-into zone.

Figure A.27 and Figure A.28 show the audit trail formats.

Fig. A.27
GST Audit Trail

```
Processing: Create GTM GST from GST masters

GST Class Description Start Eff End Eff   GST Pct AP GST Acct AR GST Acct
--------- ----------- --------- --------- -------- ----------- -----------
0         GST 0       01/01/97              0.00% 1400        2400
1         GST 1       01/01/97              5.00% 1400        2400
2         GST 2       05/29/97  12/31/99    7.00% 1400        2400
3         GST 3       07/02/97  12/31/99   10.00% 1400        2400
```

Fig. A.28
PST Audit Trail

```
Processing: Create GTM PST from PST masters
                                                                  Sums-Into
Province County City        Effective  Tax   On GST Tax Zone     Tax Zone
-------- ------ ----------- --------- ------ ------ ------------ ---------
ALB              Calgary     01/01/96  0.00%   No    ALB-Calgary  ALB
BC               Vancouver   01/01/96  7.00%   Yes   BC-Vancouv   BC
NFL              Bay Roberts 01/01/96  8.00%   No    NFL-BayRob   NFL
ONT              Bala        01/01/96  8.00%   Yes   ONT-Bala     ONT
PEI              Souris      01/01/96  10.00%  Yes   PEI-Souris   PEI
```

Troubleshooting GTM Setup

After you run CAN to GTM–Setup, verify that the GTM setup is correct before you continue with the conversion. The errors listed in Table A.37 can cause issues or unexpected values. Before proceeding to the master conversion, review the audit trail, the GTM reports for the new records, and Global Tax Management Control settings, and correct any errors.

Subsequent setups do not automatically overwrite records that previous ones created. To set up new records, first delete the old ones. If you rerun the setup, remove the records the earlier setup attempt created. Rerun the setup with Delete Previous GTM set to Yes and Convert Tax Masters set to Yes.

Note If you must rerun the setup after you have run any of the other conversion programs, restore the database first. Then, rerun the setup and any other conversion programs you ran previously. This action is necessary to perpetuate changes to master data, transactions, and tax details.

Table A.37
Troubleshooting the GTM Setup

Error	Explanation
Tax system must be Canadian.	The CAN to GTM setup can only be run on a Canadian tax system.
Must delete previous GTM when converting.	When you set Convert Tax Masters to Yes, also set Delete Previous GTM to Yes.
On GST has changed, cannot convert prior to this date.	The setup can only convert the current tax environment, not previous variations. If a city's GST Only status changed in the span of time included in the conversion, the setup creates tax environments only for current conditions.

Warning Do not correct records programmatically. This approach often causes additional problems.

How the Setup Changes GTM Records

The following is a technical description of how the setup updates the database.

The menu-level program for CAN to GTM–Setup is `txcancnv.p`. This program calls subprograms (primarily `txcantax.p`) that set the GTM tax values in the individual database tables.

Table A.38 lists the affected database tables and summarizes the changes.

Table A.38
Changes to GTM Records

Tables	Summary of Changes
Address Master (ad_mstr)	`txcantax.p` creates one ~taxes record for the database. It also scans si_mstr and creates an address record for any company site that does not already have one.
Generalized Code Master (code_mstr)	For each vt_mstr record, `txcantax.p` creates tax classes for GST only and GST + PST. If a class file is referenced, it creates the specified tax classes and tax usages, if any. For the GST rate in vt_mstr and each PST rate in tax_mstr, `txcantax.p` creates a tax type code. For non-taxable transactions, it also creates a default NON-TAX tax type. Finally, for each ship-to tax zone, `txcantax.p` creates a tax environment and assigns it the tax types associated with the tax zone.
Country Master (ctry_mstr)	`txcantax.p` creates a record for the default country code specified in the selection data.
Address List Detail (ls_mstr)	`txcantax.p` creates ls_mstr record for each new ad_mstr record, if any.
Tax Master (tx2_mstr)	For each vt_mstr record, `txcantax.p` creates a tax rate for GST only and GST + PST. For each tax_mstr record, `txcantax.p` creates multiple tx2_mstr records, one for each combination of (1) GST only and GST + PST and (2) each GST class. (For example, if there are three GST classes, the setup generates six tx2_mstr records.) It also runs `txtx2_nt.i` to create a non-taxable tax rate and `txtxmeth.i` to create tax method 01.
Tax Control (txc_ctrl)	`txcantax.p` sets txc_ctry_code and txc_tax_code from the selection data. It sets txc_method to 01, txc_by_line to Yes, txc_inv_disc and txc_pmt_disc to No, and txc_rcpt_tax_point to Yes.
Tax Base Detail (txbd_det)	`txcantax.p` creates a tax base record for PST + GST.
Tax Environment Master (txe_mstr)	`txcantax.p` creates tax environment zone detail records for every tax environment code it generates for code_mstr.
Tax Environment Detail (txed_det)	`txcantax.p` creates tax environment tax type detail records for every tax environment code it generates for code_mstr.
Tax Zone Master (txz_mstr)	`txcantax.p` creates a top level sums-into tax zone for the new ctry_mstr record. For each province/county/city combination in tax_mstr, it creates a ship-to tax zone.

Converting Master Records

Once you finish the GTM setup, the next activity is to update tax settings in the following master records:

- Suppliers
- Customers
- Product Lines
- Items
- Trailer Codes
- Service Categories

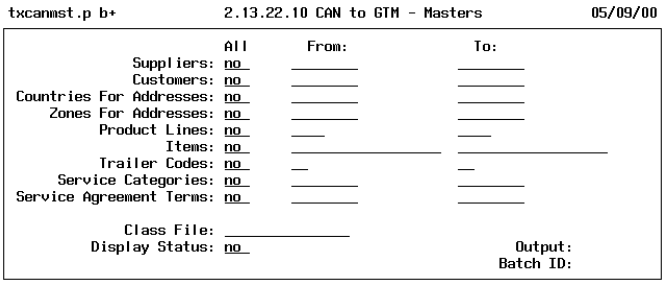
- Service Agreement Terms

GTM has additional fields and could require new values for existing fields.

CAN to GTM–Masters

Use CAN to GTM–Masters (2.13.22.10) to convert master records.

Fig. A.29
CAN to GTM–Masters (2.13.22.10)



In addition to updating the tax settings in the master records, this program assigns tax zone codes to supplier, customer, and company address records. For verification of changes, the program generates an audit trail.

Important Before you run CAN to GTM–Masters, do the following:

- Run CAN to GTM–Setup (2.13.22.9).
- To avoid record contention conflicts with other users, shut down the database and restart it when no other users are on the system.

You can convert all records, a range of records, or individual records. This program converts records in the order in which their selection options display on the screen. For separate audit trail reports, run the report separately for each type of record.

If you created a class file during the setup step, specify its name in the Class File field. Display Status, Output, and Batch ID are the same as in the setup program.

See “Defining Custom Tax Class and Tax Usage Codes” on page 121.

Master Audit Trail

CAN to GTM–Master prints a report of changed records. The format varies depending on the records included in the conversion. For each group of converted records, the report shows the record number and name and the before and after tax information. The tax information includes the country code, tax zone code, GST and PST taxable status, whether tax is included in item amounts, tax class, and tax usage.

Groups of converted records print in the same order as the screen selection criteria with a page break separating each group. Warning and error messages identify potential conversion issues.

Figure A.30 and Figure A.31 show representative audit trail formats.

Fig. A.30
Customers Audit Trail

```
Processing: Customers

Address  Name                                    GST PST Tax In TxC TaxUsage
-------- ----------------------------            --- --- ------ --- --------
10000000 Harris Steel                 Before     No  No  No     0
                                      After      No      No     0
10000001 Computer Services            Before     Yes Yes No     1
                                      After      Yes     No     1P
10000002 Niagara Byteworks            Before     Yes Yes No     2
                                      After      Yes     No     2P
```

Fig. A.31
Countries for Addresses Audit Trail

```
Processing: Countries For Addresses

Address  Name                                    Ctry Country
-------- ----------------------------            ---- -------
10000000 Harris Steel                 Before          Canada
                                      After      CAN  Canada
10000001 Computer Services            Before          Canada
                                      After      CAN  Canada
10000002 Niagara Byteworks            Before          Canada
                                      After      CAN  Canada
```

Troubleshooting Master Conversions

The error messages in the audit trail identify conditions you analyze and address before you convert transactions. Table A.39 lists some common errors along with explanations. Before you make corrections, restore the database from backup.

Warning Do not proceed to the transaction conversion until the master conversion audit trail is free of errors.

Table A.39
Troubleshooting the Master Conversion

Error	Explanation
Tax class cannot be converted.	Class file does not contain a GST class that matches the one in the master record.
GST class must exist 0%.	GST class master must have at least one GST class with a zero percentage.
Blank tax class not allowed.	GST class is blank in the class file.
Tax class cannot exceed 3 characters (*xxx*).	GST class in the class file is longer than three characters. The message shows the first three characters.
Tax class does not exist (*x*).	GST class in the class file does not exist in the GST master.
Tax class is not unique (*x*).	GST class occurs in multiple places in the class file.
Tax class does not exist (*xxx*).	GTM tax class in the class file does not exist in the GTM tax class master.
Tax usage cannot exceed 8 characters (*xxxxxxxx*).	GTM tax usage in the class file is longer than eight characters. The message shows the first eight characters.
Tax usage does not exist (*xxxxxxxx*).	GTM tax usage in the class file does not exist in the GTM tax usage master.
Tax class/tax usage combination is not unique (*xxx xxxxxxxx*).	GTM tax class and tax usage combination occurs in multiple places in the class file.
Conversion will ignore tax class (*x*).	Class file does not have an entry for a GST class that is in the class master.
x, *xxx*, and *xxxxxxxx* are placeholders for the actual codes displayed in the error message.	

How the Conversion Changes Master Records

The following is a detailed description of how the master conversion updates the database.

The menu-level program for CAN to GTM–Masters, `txcanmst.p`, calls subprograms that set the GTM tax values in the individual tables.

When setting the GTM tax class value, these programs can set the existing tax class code or retrieve an alternate value from a class file. See “Implementing GTM” on page 119.

Table A.40 lists the affected database tables and summarizes the changes.

Table A.40
Changes to Master Records

Tables	Summary of Changes
Address Master (ad_mstr)	In supplier records, `txcanvd.p` sets ad_taxable from vd_taxable. It also sets ad_taxc and ad_tax_usage from ad_taxc or from the class file, if any. If vd_taxable is No, ad_taxc is GST only; otherwise it is GST + PST. In customer records, `txcancm.p` sets ad_taxable from cm_taxable and ad_tax_in from cm_tax_in. It also sets cm_taxc and ad_tax_usage from cm_taxc or from the class file, if any. If cm_pst is No, the tax class is GST only; otherwise it is GST + PST. In all address records, `txcanct.p` sets ad_ctry from ad_country and the other way around. `txcanzn.p` calls `txtxzget.p` to set ad_tax_zone.
Service Category Master (fsc_mstr)	`txcanfsc.p` sets fsc_taxc from fsc_taxc or from the class file if any. Tax class is set to GST only.
Product Line Master (pl_mstr)	`txcanpl.p` sets pl_taxc from pl_taxc or from the class file, if any. If pl_pst is No, the tax class is GST only; if it is Yes, to GST + PST.
Item Master (pt_mstr)	`txcanpt.p` sets pt_taxc from pt_taxc or from the class file, if any. Tax class is set to GST only.
Service Agreement Terms and Conditions Master (sv_mstr)	`txcansv.p` sets sv_taxc and sv_tax_usage from sv_taxc or from the class file, if any. Tax class is set to GST only.
Trailer Master (trl_mstr)	`txcantrl.p` sets trl_taxc from trl_taxc or from the class file, if any. It also sets trl_taxable to Yes. If trl_pst is No, the tax class is GST only; if it is Yes, to GST + PST.

Converting Transaction Records

Once you finish converting master records, you can convert the following transaction records:

- Purchase orders and receipts
- Accounts payable vouchers and payments
- Service contracts, calls, and return material authorizations
- Sales quotes and orders
- Accounts receivable memos, invoices, and payments

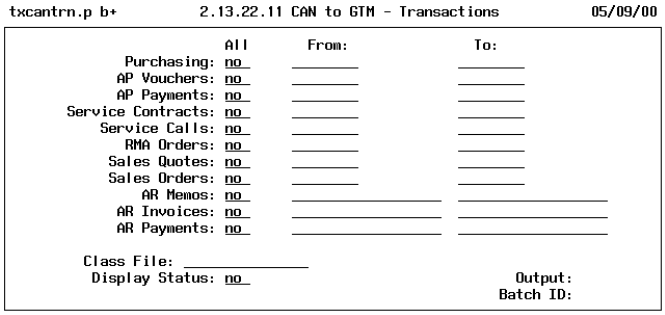
In GTM, every transaction subject to tax has a transaction tax detail record. This record stores the information used to calculate tax.

Note If you already have GTM transaction records in the database when you perform transaction conversion, the conversion process updates them using the current default tax values. Values replaced in these records include tax environment, class, usage, and so on.

CAN to GTM–Transactions

Use CAN to GTM–Transactions (2.13.22.11) to convert existing transaction records.

Fig. A.32
CAN to GTM– Transactions (2.13.22.11)



In addition to updating transactions, this program generates an audit trail for verification of changes.

Important Before you run CAN to GTM–Transactions, do the following:

- Run CAN to GTM–Setup (2.13.22.9) and CAN to GTM–Masters (2.13.22.10).
- To avoid record contention conflicts with other users, shut down the database and restart it when no other users are on the system.

You can convert all records, a range of records, or individual records. This program converts records in the same order they display on the screen.

Note For separate audit trail reports, run the report separately for each type of record.

In some cases, the record sequence is important. Purchasing transactions are converted before accounts payable vouchers and vouchers before payments. Accounts receivable memos and invoices are converted before payments.

If you created a class file during the setup step, specify its name in the Class File field. Display status, output, and Batch ID are the same as in the setup program.

See “Defining Custom Tax Class and Tax Usage Codes” on page 121.

Transaction Audit Trail

The transaction conversion prints a report of changed records. The format varies depending on the records included in the conversion. For each group of converted records, the report shows the transaction number and name and then the before and after tax information for each line item. The tax information includes the taxable status, tax environment, tax class, and tax usage. Groups of converted records print in the same order as the screen selection criteria with a page break separating each group.

Warning and error messages identify potential conversion issues. Messages that appear at the end of a transaction apply to the entire transaction; messages that appear between the Before and After line apply only to that line.

Figure A.33 and Figure A.34 show representative audit trail formats.

Fig. A.33
Accounts Payable Audit Trail

```
Processing: AP Vouchers

Ref Type Supplier Ln       GST TxC TaxUsage Tax Env vod_type vod_tax vod_tax_at
--- ---- -------- --       --- --- -------- ------- -------- ------- ----------
105 VO   32487432    Before No
                     After  Yes 0P           PEI-GG
                   1 Before No  1                    R               1
                     After  Yes 1            PEI-GG  R               Yes
                   2 Before No                                 1
                     After  No                         T       t       No
```

Fig. A.34
Service Calls Audit Trail

```
Processing: Service Calls

                          Cust
Call ID Call/SR Cust      PST   Ln Record            GST PST TxC TaxUsage Tax Env
------- ------- --------- ----- -- ---------        --- --- --- -------- -------
CA124           10000000 Yes    1 Call      Before   Yes     1
                                            After    Yes     1P           NB
CA125           10000001 Yes    1 Item      Before   Yes Yes 1
                                            After    Yes     1P           NB
CA126           10000002 Yes    1 Billing   Before   Yes Yes 1
                                            After    Yes     1P           NB
                                2 Billing   Before   No  Yes 0
                                            After    Yes     1P           NB
```

Troubleshooting Transaction Conversions

The error messages in the audit trail identify conditions you analyze and address before you resume live GTM processing. Table A.41 lists some common errors along with explanations. Before you make corrections, restore the database from backup.

Warning Do not resume live processing until the transaction conversion audit trail is free of errors. Do not correct transaction records programmatically. This approach often causes additional problems.

In addition to examining the audit trail, review the Tax Detail by Transaction Report (2.13.15.3). This report shows the tax environments, tax types, and tax amounts for the converted records. Verify that the tax calculations are as expected.

Converted transactions can have minor differences in before/after tax amounts. These differences occur because GTM uses a different calculation algorithm or rounding method than your current system. To synchronize the general ledger with the converted transactions, record adjusting entries.

Table A.41
Troubleshooting the Transaction Conversion

Error	Explanation
GST class must exist 0%.	The GST class master must have at least one GST class with a zero percentage.
Freight, brokerage, or duty charges cannot be converted.	There is no equivalent ability in GTM.
Detail tax environment must match header.	In accounts payable vouchers and accounts receivable debit/credit memos, the tax environment are the same in the header and detail lines.

To eliminate ambiguity, the audit trail shows before and after values for purchasing, accounts payable, and accounts receivable records by their Progress database field name, not their screen label. For example, the audit trail displays voucher line types in the vod_type column.

As you interpret the audit trail for the transaction conversion, it can be helpful to reference Table A.42 through Table A.46. These tables summarize the nature of before/after tax values.

Note In these tables, quotation marks indicate a value that cannot be translated.

Table A.42
CAN to GTM, Purchasing Transactions

Status	Tax System	pod_taxable	prh_tax_at
Taxable	CAN	Yes	Tax class
	GTM	Yes	"Yes"
Non-taxable	CAN	No	"0"
	GTM	Yes	"Yes"

Table A.43
CAN to GTM, AP Voucher Receiver Lines

Status	Amt	Tax System	GST	TxC	vod_type	vod_tax	vod_tax_at
Taxable	Item	CAN	Yes	GST class	"R"	Blank	GST class
		GTM	Yes	Tax class	"R"	Blank	"Yes"
	Tax	CAN	No	Blank	Blank	GST class	Blank
		GTM	No	Blank	"T"	"t"	"No"
Non-taxable	Item	CAN	No	Blank	"R"	Blank	0% GST class
		GTM	Yes	Tax class	"R"	Blank	Blank
	Tax[1]	CAN	No	Blank	Blank	GST class	Blank
		GTM	–	–	–	–	–

1. The conversion deletes Canadian tax lines resulting from non-taxable amounts.

Table A.44
CAN to GTM, AP Voucher Memo Lines

Status	Amt	Tax System	GST	TxC	vod_type	vod_tax	vod_tax_at
Taxable	Item	CAN	Yes	Blank	Blank	Blank	GST class
		GTM	Yes	Tax class	Blank	Blank	"Yes"
	Tax	CAN	No	Blank	Blank	GST class	Blank
		GTM	No	Blank	"T"	"t"	"No"

Status	Amt	Tax System	GST	TxC	vod_type	vod_tax	vod_tax_at
Non-taxable	Item	CAN	No	Blank	Blank	Blank	0% GST class
		GTM	Yes	Tax class	Blank	Blank	"No"
	Tax[1]	CAN	No	Blank	Blank	GST class	Blank
		GTM	–	–	–	–	–

1. The conversion deletes Canadian tax lines resulting from non-taxable amounts.

Table A.45
CAN to GTM, AR Invoices

Status	Amt	Tax System	ard_tax	ard_tax_at	ard_taxc
Taxable	Item	CAN	Blank	GST class	Blank
		GTM	Blank	Tax class	Tax class
	Tax	CAN	GST class or blank[1]	Blank	Blank
		GTM	"t"	"No"	Blank
Non-taxable	Item	CAN	Blank	0% GST class	Blank
		GTM	Blank	Tax class	Tax class
	Tax[2]	CAN	0% GST class	Blank	Blank
		GTM	–	–	–

1. Blank for PST.
2. The conversion deletes Canadian tax lines resulting from non-taxable amounts.

Table A.46
CAN to GTM, AR DR/CR Memos

Status	Amt	Tax System	ard_tax	ard_tax_at	ard_taxc
Taxable	Item	CAN	Blank	GST class	Blank
		GTM	Blank	"Yes"	Tax class
	Tax	CAN	GST class[1]	Blank	Blank
		GTM	"t"	"No"	Blank
Non-taxable	Item	CAN	Blank	0% GST class	Blank
		GTM	Blank	"No"	Tax class
	Tax[2]	CAN	–	–	–
		GTM	–	–	–

1. DR/CR memos do not calculate PST.
2. The conversion deletes Canadian tax lines resulting from non-taxable amounts.

How the Conversion Changes Transaction Records

The following is a technical description of how the transaction conversion updates the database.

The menu-level program for CAN to GTM–Transactions is `txcantrn.p`. This program calls subprograms that set the GTM tax values in the individual database tables. For all transactions, the conversion also generates corresponding tax detail records in the Tax Detail (tx2d_det) table.

When setting the GTM tax class value, these programs set the existing tax class code or retrieve an alternate value from a class file. See “Defining Custom Tax Class and Tax Usage Codes” on page 121.

Table A.47 lists the affected database tables and summarizes the changes.

Table A.47
Changes to Transaction Records

Tables	Summary of Changes
Accounts Receivable Detail (ard_det)	For debit/credit memos, `txcanarm.p` sets ard_tax and ard_tax_at. It also sets ard_taxc (GST only), ard_tax_usage, and ard_tax_at.
Accounts Receivable Master (ar_mstr)	For debit/credit memos, `txcanarm.p` sets ar_tax_env.
Service/Support Call Master (ca_mstr)	`txcanca.p` sets ca_taxc and ca_tax_usage from ca_taxc or from the class file, if any. If cm_pst is No, ca_taxc is GST only; otherwise, it is GST + PST. It also sets ca_tax_env using `txtxeget.p`.
Invoice History Detail (idh_hist)	`txcanari.p` sets idh_taxc and idh_tax_usage from idh_taxc or from the class file, if any. It sets idh_taxable to Yes. If idh_pst is No, idh_taxc is GST only; otherwise, it is GST + PST. It also sets idh_tax_env using `txtxeget.p`.
Invoice History Master (ih_hist)	`txcanari.p` sets ih_taxable to Yes and ih_pst_pct to 0. It sets ih_taxc (GST only) and ih_tax_usage from ih_taxc or from the class file, if any. It also sets ih_tax_env using `txtxeget.p`.
Service/Support Call Item Detail (itm_det)	`txcanca.p` sets itm_taxc (GST + PST) and itm_tax_usage from itm_taxc or from the class file, if any. It sets itm_taxable to Yes. It also sets itm_tax_env using `txtxeget.p`.
Purchase Order Detail (pod_det)	`txcanpo.p` sets pod_taxable to Yes and pod_tax_env using `txtxeget.p`. If pod_pst is No, then pod_taxc is 0; otherwise, pod_taxc is 0P.
Purchase Order Master (po_mstr)	`txcanpo.p` sets po_tax_pct[1], po_tax_pct[2], and po_tax_pct[3] to 0. It sets po_taxable to Yes. It sets po_taxc and po_tax_usage from the ad_taxc value for the corresponding supplier. It also sets po_tax_env using `txtxeget.p`.
Purchase Order Receipt History (prh_hist)	`txcanpo.p` sets prh_taxc from pod_taxc, prh_tax_usage from pod_tax_usage, and prh_tax_env from pod_tax_env. Since po_taxable is Yes, it sets prh_tax_at to Yes.
Sales Quotation Detail (qod_det)	`txcanqo.p` sets qod_taxable to Yes. It sets qod_taxc and qod_tax_usage from qod_taxc or from the class file, if any. It sets qod_taxable to Yes. If qod_pst is No, tax class is GST only; otherwise, it is GST + PST. It also sets qod_tax_env using `txtxeget.p`.
Sales Quotation Master (qo_mstr)	`txcanqo.p` sets qo_tax_pct[1], qo_tax_pct[2], and qo_tax_pct[3] to 0. It sets qo_taxable to Yes and qo_pst_pct to 0. It sets qo_taxc and qo_tax_usage from qo_taxc or from the class file, if any. If cm_pst is No, tax class is GST only; otherwise, it is GST + PST. It also sets qo_tax_env using `txtxeget.p`.
Return Material Authorization Master (rma_mstr)	`txcanrma.p` sets rma_taxc from rma_taxc or from the class file, if any. It sets rma_taxable to Yes. If cm_pst is No, tax class is GST only; otherwise, it is GST + PST.
Service Contract Detail (sad_det)	`txcansc.p` sets sad_taxc and sad_tax_usage from sad_taxc or from the class file, if any. It sets sad_taxable to Yes. If sad_pst is No or sa_prefix is QA, tax class is GST only; otherwise, it is GST + PST. It also sets sad_tax_env from sa_site and sa_taxc using `txtxeget.p`.

Table A.47 — *Changes to Transaction Records* (Page 1 of 2)

Tables	Summary of Changes
Service Contract Master (sa_mstr)	`txcansc.p` sets sa_tax_pct[1], sa_tax_pct[2], and sa_tax_pct[3] to 0. It sets sa_taxable to Yes and sa_can_tax to 0. It sets sa_taxc and sa_tax_usage from sa_taxc or from the class file, if any. If cm_pst is No, tax class is GST only; otherwise, it is GST + PST. It also sets sa_tax_env using `txtxeget.p`.
Service/Support Billing Detail (sfb_det)	`txcanca.p` sets sfb_taxc from itm_taxc and sfb_tax_usage from itm_tax_usage. It sets sfb_taxable to Yes. It also sets sfb_tax_env using `txtxeget.p`.
Sales Order Detail (sod_det)	`txcanso.p` sets sod_taxc and sod_tax_usage from sod_taxc or from the class file if any. It sets sod_taxable to Yes. If sod_pst is No, tax class is GST only; otherwise, it is GST + PST. It also sets sod_tax_env using `txtxeget.p`.
Sales Order Master (so_mstr)	`txcanso.p` sets so_taxable to Yes and so_pst_pct to 0. It sets so_tax_pct[1], so_tax_pct[2], and so_tax_pct[3] to 0. It sets so_taxc and so_tax_usage from so_taxc or from the class file, if any. If cm_pst is No, tax class is GST only; otherwise, it is GST + PST. It also sets so_tax_env using `txtxeget.p`.
Tax Detail (tx2d_det)	`txcanpo.p` creates tax details for purchase orders (GTM transaction type 20), receivers (21), reconciliations (23), and returns (25). `txcanapv.p` creates tax details for vouchers (22) and recurring vouchers (32). `txcanapp.p` creates tax details for accounts payable tax on discount at payment (29). `txcanqo.p` creates tax details for sales quotes (10). `txcanso.p` creates tax details for invoiced service calls (38), return material authorizations (36), sales orders (11), and pending invoices (13). `txcanarm.p` creates tax details for debit/credit memos (18). `txcanari.p` creates tax details for invoices (16). `txcanarp.p` creates tax details for accounts receivable tax on discount at payment (19). `txcansc.p` creates tax details for service quotes (33) and service contracts (34).
Voucher Detail (vod_det)	`txcanapv.p` sets vod_taxable to Yes. It sets vod_taxc (GST only) and vod_tax_usage from vod_tax_at. It sets vod_tax, vod_tax_at, and vod_type. It also sets vod_tax_env using `txtxeget.p`.
Voucher Master (vo_mstr)	For vouchers, `txcanapv.p` sets vo_tax_pct[1], vo_tax_pct[2], and vo_tax_pct[3] to 0. It sets vo_taxable, vo_taxc, and vo_tax_usage. It also sets vo_tax_env using `txtxeget.p`.

Table A.47 — *Changes to Transaction Records* (Page 2 of 2)

Appendix B

Running the Fixed Assets Migration Utility

This appendix describes how to run the Fixed Assets Migration utility.

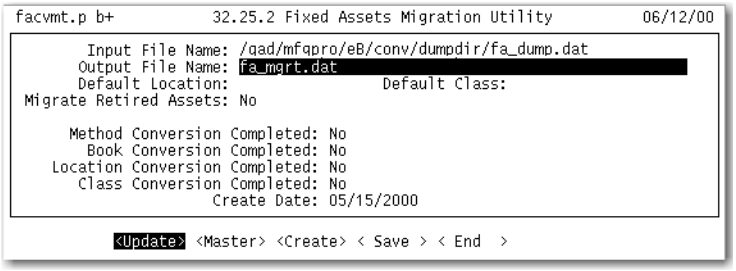
Running the Fixed Assets Migration Utility

The following sections describe the Fixed Assets Migration Utility (32.25.2). Use this information to map the legacy data model to the enhanced model and to resolve discrepancies. When ready, select the Create option on the Fixed Assets Migration Utility screen and press Enter to load the data into the enhanced Fixed Assets module.

The Fixed Assets Migration Utility tracks migration process milestones. It also lets you map the individual migration data types—methods, books, locations, and classes—in stages and then save these intermediate stages to a migration file. After you have mapped all of the legacy data to the enhanced data model, select the Create option and press Enter. This action loads the data from the migration file into the enhanced Fixed Assets module.

Important This update can only be done once.

Fig. B.1
Fixed Assets Migration Utility (32.25.2)



Most of the fields in the Fixed Asset Migration Utility header cannot be updated. They display current mappings set in detail menus and screens. Use the options at the bottom along with the Enter key to access the detail menus and screens. The following sections describe these features in detail.

Options

Update. Use this option to update the Output File Name, Default Location, Default Class, and the Migrate Retired Assets field.

Master. Used to access the Master Files Migration screen.

Create. Use this option to load legacy data into the enhanced Fixed Assets module after all the data is mapped to the enhanced model. You can only run this function once.

Save. Used to save the completed work to the file specified in the Output File Name field.

End. Use this option to end a migration utility session. All work is saved to the file specified in the Output File Name field.

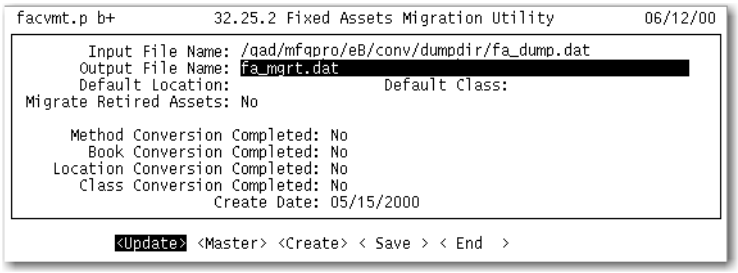
Setting Migration Defaults

Choose Update to set up the default parameters for your migration.

Use the Default Location and Default Class fields to specify a legacy location and class to default for legacy assets that do not already have this information. The location and class must be part of the legacy data. You can run the Fixed Assets Migration Report (32.25.3) to generate a list of your legacy classes and locations.

Set the Migrate Retired Assets field to Yes to migrate retired legacy asset data. Set the field to No to exclude retired assets from the migration.

Fig. B.2
Fixed Assets Migration Utility (32.25.2)



1 The Input File Name field defaults to `fa_dump.dat`. Enter only the file name. If you specified a different file name in the MFG/CONV environment prompts, specify the file name for your legacy data dump file. Press Enter to continue.

2 Select the Update option and press Enter.

3 In the Output File Name field, specify the full path, including the file name, to the location where you want to store your migration file. Use this file to store your work as you map the legacy data to the new fixed assets model. The default file name is `fa_mgrt.dat`.

4 In the Default Location field, enter a default legacy location ID. Fixed asset location IDs refer to the accounting location of the fixed asset. This location does not have to be the physical location of the asset.

5 In the Default Class field, enter a default legacy class ID.

Classes group similar fixed assets together and define:

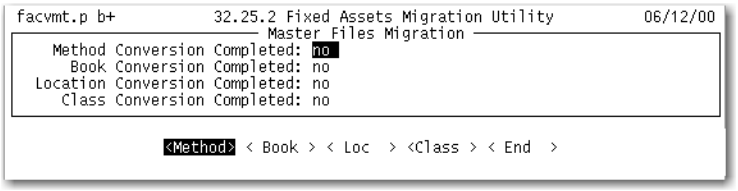
- GL accounts
- Depreciation books
- Service lives for calculating depreciation
- Depreciation methods

6 In the Migrate Retired Assets field, enter Yes to migrate retired assets to the new system. Enter No if you do not want to migrate retired assets.

Mapping Legacy Data

Migrating legacy data to the enhanced module requires mapping the existing data models to the new models. Use the Master Files Migration screen to monitor the conversion of the legacy data models. Master Files Migration also provides access to the maintenance programs for each data model. These programs let you map legacy models to enhanced models.

Fig. B.3
Master Files Migration



Access the maintenance program for each data model using the options at the bottom of Master Files Migration along with the Enter key.

Options

Method. Used to access Method Migration.

Book. Use this option to access Book Migration.

Loc. Used to access Location Migration.

Class. This option accesses Class Migration.

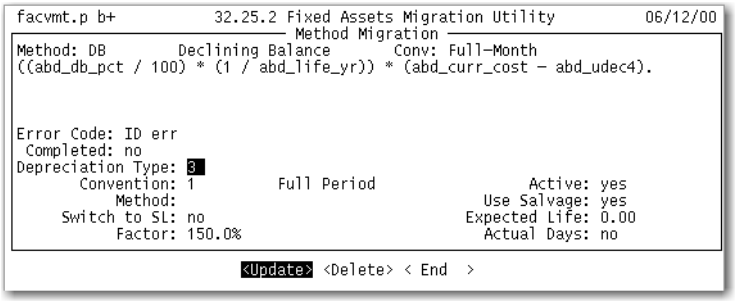
End. Use this option to end a Master Files Migration session and return to the Fixed Asset Migration utility.

Conversion Methods

Use Method Migration to convert legacy depreciation methods to the depreciation methods supplied with the enhanced Fixed Assets module.

To convert legacy methods, use the following figure and instructions.

Fig. B.4
Method Migration



1 Use the arrow keys to scroll through the legacy methods. The legacy method, description, convention, and equation display.

2 If an asset does not use this method, select the Delete option and press Enter to remove it. The system verifies that the method is not used and prompts to confirm the deletion.

3 To convert the legacy method, select the Update option and press Enter.

4 In the Depreciation Type field, select the enhanced depreciation method that most closely resembles the legacy method. The following depreciation methods are supplied with the enhanced module:

- 1: Straight Line
- 2: Units of Production
- 3: Declining Balance
- 4: Sum of the Years' Digits
- 5: Flat Rate
- 6: Custom Table

5 To modify the standard depreciation methods supplied with the enhanced Fixed Assets module, complete the following fields:

- Convention
- Method
- Switch to SL
- Factor
- Active
- Use Salvage
- Expected Life
- Actual Days

Note If you are using the custom table depreciation method, define a custom table in Method Maintenance (32.1.1) before mapping it to a legacy method. Use the same method ID from Method Maintenance for the method in Method Migration.

6 If an error exists with the new method, the Error Code field displays one of the following tables:

- ID_err: An error exists with the new method ID. For example, the method ID is missing or there is a duplicate method ID.
- type_err: An error exists with the new method description. For example, the enhanced module does not support the method.
- conv_err: An error exists with the new convention. For example, the convention is missing or the convention is not compatible with the method.
- table_err: An error exists with the new table method. The following are examples of possible errors:
  - The corresponding method defined in Method Maintenance is not a custom-table method.
  - There is a discrepancy with the estimated life.
  - Table detail is undefined in Method Maintenance.
- life_err: An error exists with the new life. For example, the method is a custom table and the new life is zero.

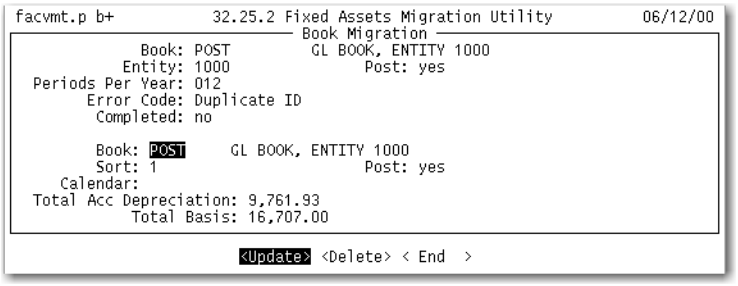
7 After you correct any errors, the Completed field updates to Yes for the converted method.

8 Repeat steps 1 through 7 for each legacy method.

Converting Books

Use Book Migration to convert legacy books to books used in the enhanced Fixed Assets module. To convert legacy books, use the following figure and instructions. In the legacy system, asset cost is associated with depreciation books. Therefore, if an asset contains multiple books, the asset cost comes from the posting book.

Fig. B.5
Book Migration



1 Use the arrow keys to scroll through the legacy books. The legacy book ID, description, Entity, and Post field display.

2 If no assets use this book, select the Delete option and press Enter to remove it. The system verifies that the book is not used and prompts to confirm the deletion.

Note Any legacy asset books that use a depreciation type of None are not created in the new fixed asset system.

3 To convert the legacy book, select the Update option and press Enter.

4 Complete the following fields:

- Book
- Description
- Sort Code
- Post
- Calendar

Note You can have only one posting book in the system. If the book was not a posting book in the legacy system, you cannot change it to a posting book for the new system.

5 Total Acc Depreciation and Total Basis are automatically calculated. Total Acc Depreciation displays the total accumulated depreciation for the legacy book. Total Basis displays the total basis for all of the fixed assets for the legacy book.

6 If duplicate book IDs exist, the Error Code field displays Duplicate ID.

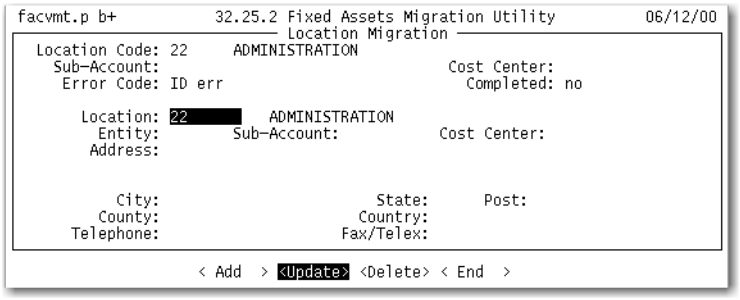
7 After you correct any errors, the completed field updates to Yes for the converted book.

8 Repeat steps 1 through 7 for each legacy book.

Converting Locations

Use Location Migration to convert legacy locations to locations used in the enhanced Fixed Assets module. To convert legacy locations, use the following figure and instructions.

Fig. B.6
Location Migration

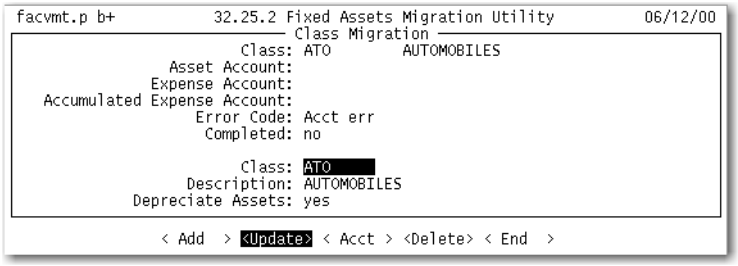


1. Use the arrow keys to scroll through the legacy locations. The legacy Location Code, Description, Sub-Account, and Cost Center display. For each legacy location, the location ID automatically fills in with the legacy location ID.
2. If an asset does not use this location, select the Delete option and press Enter to remove it. The system verifies that the location is not used and prompts to confirm the deletion.
3. To convert the legacy location, select the Update option and press Enter.
4. Complete the following fields:
   - Location ID
   - Description
   - Entity
   - Sub-Account
   - Cost Center
   - Address
   - Telephone
5. If an error exists with the new location, the Error Code field displays one of the following codes:
   - ID err: An error exists with the new ID. For example, another module uses the location ID and you must set up a new ID.
   - en err: An error exists with the new entity. For example, the entity code is undefined in Entity Code Maintenance (25.3.1).
   - sub err: An error exists with the new sub-account. For example, the sub-account is undefined in Sub-Account Code Maintenance (25.3.17).
   - cc err: An error exists with the new cost center. For example, the cost center is undefined in Cost Center Code Maintenance (25.3.20).
6. After you correct any errors, the Completed field updates to Yes for the converted location.
7. Repeat steps 1 through 6 for each legacy location.

Converting Classes

Use Class Migration to convert legacy classes to classes used in the enhanced Fixed Assets module and set up default GL accounts. To convert legacy classes, refer to the following figure and instructions.

Fig. B.7
Class Migration



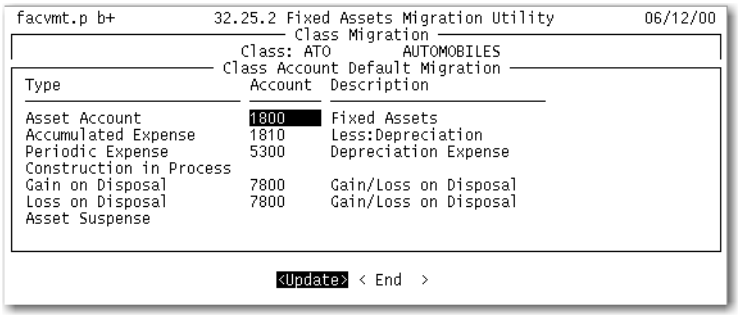
1 Use the arrow keys to scroll through the legacy classes. The legacy class ID, description, Asset Account, accumulated depreciation account, and depreciation expense account display. For each legacy class, the class ID and description are automatically filled with the legacy data.

2 If an asset does not use this class, select the Delete option and press Enter to remove it. The system verifies that the class is not used and prompts to confirm the deletion.

3 To convert the legacy class, choose Update and enter whether this fixed-asset class is depreciated.

4 Select the Acct option and press Enter to modify the default GL accounts for the fixed-asset class.

Fig. B.8
Class Account Default Migration



5 For each class, specify a GL account for the following accounts:

- Asset Account
- Accumulated Expense
- Periodic Expense
- Construction In Process
- Gain on Disposal
- Loss on Disposal
- Asset Suspense

6 If any fixed-asset accounts are undefined, the Error Code field displays acct err.

7 After you correct any errors, the Completed field updates to Yes for the converted class.

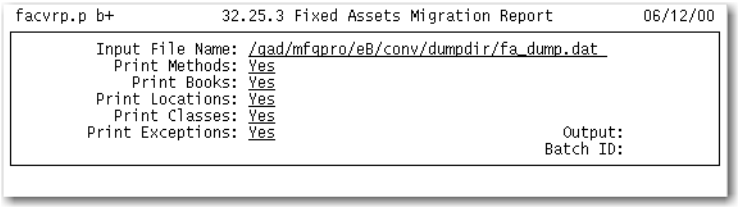
8 Repeat steps 1 through 7 for each legacy class.

Migration Reporting

Use Fixed Assets Migration Report (32.25.3) to generate a report that provides a summary of the migrated fixed-asset data. The report shows the migrated data from the legacy system and the newly mapped data for the new system.

You can choose to include books, methods, locations, classes, and exceptions in the report.

Fig. B.9
Fixed Assets Migration Report (32.25.3)



1. In the Print Books field, enter Yes to include migrated book data from the legacy system and the newly mapped book data in the report. Enter No to exclude book data.
2. In the Print Methods field, enter Yes to include migrated method data from the legacy system and the newly mapped method data in the report. Enter No to exclude method data.
3. In the Print Locations field, enter Yes to include migrated location data from the legacy system and the newly mapped location data in the report. Enter No to exclude location data.
4. In the Print Classes field, enter Yes to include migrated class data from the legacy system and the newly mapped class data in the report. Enter No to exclude class data.
5. In the Print Exceptions field, enter Yes to include exceptions for your data. Enter No to exclude the exceptions.

Appendix C

Converted Data

This appendix describes how selected data is converted.

Overview

The conversion creates many new objects, some of which are created based on user-specified values provided in the Conversion Parameters Utility for QAD Enterprise Financials. These new objects include items, such as business relations, shared sets, profiles, invoice status codes, project status codes, and project groups.

This section describes some of the objects the conversion creates and the naming conventions the conversion uses when creating the objects.

Important This appendix only applies to conversions from a pre-Enterprise Edition version of the QAD ERP application to Enterprise Edition. It does not apply to upgrades from an Enterprise Edition installation.

Layers

The conversion creates the default accounting layers that QAD Enterprise Financials require.

Business Relations

The conversion creates a business relation and an address of type Head Office for employees and addresses of the following types:

- Customer
- Supplier
- Employee
- Carrier
- Company
- Slsprsn
- Engineer
- Our_bank (bank address for the business's operational cash)
- C/S_bank (customer and supplier bank addresses)
- Faloc (Fixed Asset location ID address)

Shared Sets

The conversion creates a number of shared sets for each of the domains in the source database.

The conversion only creates shared sets for active domains that belong to the database being converted. Each shared set is named using a combination of the source domain name and the shared set type.

The following table lists the shared sets created for each source domain.

Table C.1
Shared Sets

Shared Set Code	Shared Set Type
<domain>Cost Center	Cost Center
<domain>Customer	Customer
<domain>Daybook	Daybook
<domain>Ex Rate	Exchange Rate
<domain>Account	Account
<domain>Project	Project
<domain>SubAccount	SubAccount
<domain>Supplier	Supplier
<domain>COAMaskDiv	Sub-Account COA Mask
<domain>COAMaskCC	Cost Center COA Mask
<domain>COAMaskProj	Project COA Mask

Entities

The conversion links entities to the shared set of the domain to which the entity belongs.

Profiles

The conversion creates the default profiles listed in the following table:

Table C.2
Default Profiles

Profile Code	Profile Type
Default Bank-<domain>	Banking Entry Daybook Profile
<cost center>-<domain>-C	Cost Center Profile
<AR account>-<domain>-A	Customer Account Profile
<project>-<domain>-P	Project Profile
<Purchases account>-<domain>-PA	Purchase Account Profile
<Sales account>-<domain>-SA	Sales Account Profile
<sub-account>-<domain>-S	Sub Account Profile
<AP account>-<domain>-A	Supplier Account Profile

Accounts

The conversion does not convert Memo and Statistical type accounts or blank sub-accounts, cost centers, or projects. Instead, the conversion replaces the blank values with default values prompted for by the Conversion Parameters Utility.

Analysis Type and Analysis Limitations

The conversion uses Code Range Master (cr_mstr) to determine the analysis type setting for an account. For example, if an account code is within the range defined in Code Range Master for a cost center, the account is assigned an analysis type of Cost Center.

There are exceptions to the assignment of the analysis type setting for some account types. For example, system accounts (except for PO receipts system accounts) must have an analysis type of None. The validations in QAD Enterprise Financials define these limitations.

The other exception when using the Code Range Master to assign the analysis type setting occurs when converting from 8.6E or 9.0. In this scenario, there are no ranges defined for projects. Therefore, the pre-conversion GL Account/Project Range Utility (`uxglproj.p`) is used to identify which accounts use project analysis. For the analysis limitations, the default value is None, but there are exceptions.

Enterprise Edition accounts that use automatic posting cannot have a combined analysis type of Both and an analysis limitation of None. Automatic accounts include customer and supplier control and payment accounts, and system accounts other than PO receipt accounts. Therefore, for any accounts with a posting type of Automatic and an analysis type of Both, the conversion sets the analysis limitation to Both. All other accounts have an analysis limitation of None.

COA Mask

The conversion creates a COA mask range for every Code Range Master (cr_mstr) range in the source database. All pre-existing Account Sub-Account Cost Center Master (asc_mstr) records (combinations used in historical transactions) are deleted.

For MFG/PRO versions below eB2, the Code Range Master table is populated from sub-account and cost center combinations from the following tables as part of the conversion to QAD Enterprise Edition:

- Sub-Account/Account Validation Detail (sbd_det) table
- Cost Center Account Validation Detail (ccd1_det) table
- Cost Center Sub-Account Validation Detail (ccd2_det) table

When converting a GL account, the conversion uses the ranges in Code Range Master to determine whether the account uses cost center or project analysis. If an account:

- Does not use cost center or project analysis, the analysis type is set to None
- Only uses project analysis, the analysis type is set to Project
- Only uses cost center analysis, the analysis type is set to Cost Center
- Uses both cost center and project, the analysis type is set to Both

Note For MFG/PRO versions below 9.0, project code ranges are set to All. This means that these project codes are valid for use with all accounts that allow project analysis (as defined in the GL Account/Project Range Utility).

The conversion only assigns a default cost center to an account if its analysis type is Cost Center. Similarly, the conversion only assigns a default project to an account if its analysis type is Project.

If an account has an analysis type of Both or None, the conversion sets the account's analysis limitation to None, and the default values are left blank. However, there is one exception to this rule. In QAD Enterprise Financials, if an account has a posting type of Automatic and an analysis type of Both, set the analysis limitation to Both. Also assign the account a default cost center and project.

Conversion Accounts

The conversion creates a "conversion account" for every general ledger account that is flagged as a control account within the pre-conversion Control Account Utility (`uxctrl.p`). The accounts use the format QAD-xxxx, where xxxx is a sequential number the conversion assigns. The account description is obtained by concatenating the original account number and description.

Their usage is similar to that of "take on accounts" during a system implementation. Conversion accounts serve as temporary placeholders for Accounts Receivable and Accounts Payable general ledger balances until the sub-ledger transactions are loaded into the database or, in this instance, converted.

During the conversion, a number of different transactional conversions are performed. The first is the GL Transactions conversion. The Supplier Invoice Conversion, Supplier Payment Conversion, Customer Invoice Conversion, and Customer Payment Conversion follow this conversion.

When a transaction that references a control account is encountered during the GL Transactions conversion, the conversion replaces the control account code with its corresponding conversion account code. This replacement effectively moves the general ledger balance from the control account to the conversion account. The control account balances are reinstated during the subsequent Invoice and Payment transaction conversions. Depending on the transaction being converted (Invoice, Credit Note or Payment), the general ledger transactions created debit or credit the control account and offset the conversion account.

Ideally, all conversion accounts have a zero balance following conversion. However, in some situations, balances remain.

In most cases of a remaining balance against a conversion account, an imbalance between the sub-ledger and the general ledger existed before conversion. The conversion simply moved the imbalance from the control account to the conversion account. This action is necessary because within Enterprise Edition, the sub-ledger, and general ledger must be in balance. In this instance, you decide what to do with the conversion account balance.

In instances in which a number of similar type control accounts exist, you could find that balances remain against individual conversion accounts. However, the total of the Accounts Payable or Accounts Receivable conversion accounts has a zero balance. While there are a number of reasons why this situation could occur, a common cause is the correction of sub-ledger transactions using

Journal entries within the source database. That is, an invalid Control Account code was used for a transaction and then corrected with a Journal entry. In this case, the conversion account balances can simply be netted off using a Journal transaction and the accounts set to inactive.

Finally, balances could remain due to corrupt data within the source database. Thoroughly investigate and resolve any remaining balances against a conversion account. Once the accounts are balanced, change the status of the accounts to inactive.

GL Transactions

The conversion does not convert the existing GL transactions against the original control account, but instead creates posting lines in QAD Enterprise Financials against the special conversion accounts. This process solves a potential balance discrepancy between the converted sub-ledger (AR or AP) and the converted control account balances for the corresponding accounts (AR control and AP control). Agreement between the sub-ledgers and control accounts is a key check in QAD Enterprise Financials, and you must resolve any discrepancies during the conversion. If a GL transaction is posted to a non-control account, the transaction is converted as-is.

Unbalanced GL Transactions

A database can contain unbalanced GL transactions. A potential cause of unbalanced transactions is that the database contains transactions from other financial software that were previously converted to the QAD application.

Unbalanced transactions in a closed fiscal year are converted as-is. If the entire closed year is out of balance for an individual entity, the conversion creates a balancing transaction. The reason is that if the historical years are unbalanced, the trial balance in QAD Enterprise Financials never balance.

An accounting year is composed of all GL calendar periods for an individual GL calendar year. To determine if a balancing transaction is needed for an entity in a closed year, all converted GL transactions for an accounting year and entity are tracked. If the total does not balance, the conversion creates a new posting and posting line to balance the year and entity. The conversion creates the balancing transaction in the appropriate historical period, and uses the GL account entered in the Conversion Parameters Utility for the balancing transactions.

Note The calculations that determine if the accounting year is balanced do not include year-end postings (transactions of type YR). Year-end postings are discussed in the following text.

The conversion modifies any unbalanced transactions in an open fiscal year so that the year-end closing process in QAD Enterprise Financials can run without error. The conversion automatically creates a single posting line for the out-of-balance amount against an account specified for this purpose before starting the conversion. This process balances the unbalanced transaction. All balancing changes the conversion makes are recorded in a log file. The balancing transaction can also be easily identified by analyzing the postings made against this special account.

The conversion treats year-end transactions (type YR) differently because these transactions are always one-sided in earlier versions of the QAD ERP application. The conversion always creates a balancing posting line for year-end transactions. This balancing posting line is created in the historical period and uses the P&L (balance sheet) account from General Ledger Control (co_ctrl.co_pl). As noted previously, the year-end posting is not considered when determining if the overall year is balanced.

In European Accounting, the year-end transaction is not one-sided, but must be fully balanced. When European Accounting is used, the conversion does not create a balancing posting line for year-end transactions.

AR Transactions

The conversion posts each total invoice or payment amount to the AR account defined in the customer master. The conversion then reverses this amount from the corresponding conversion account for the AR account of the original transaction.

If the AR account used for the original transaction is not an actual AR control account provided in the pre-conversion Control Account Utility or GL Account Type Utility, the posting line is changed to use the account from one of these utilities. The conversion records changes of this type in a separate log file.

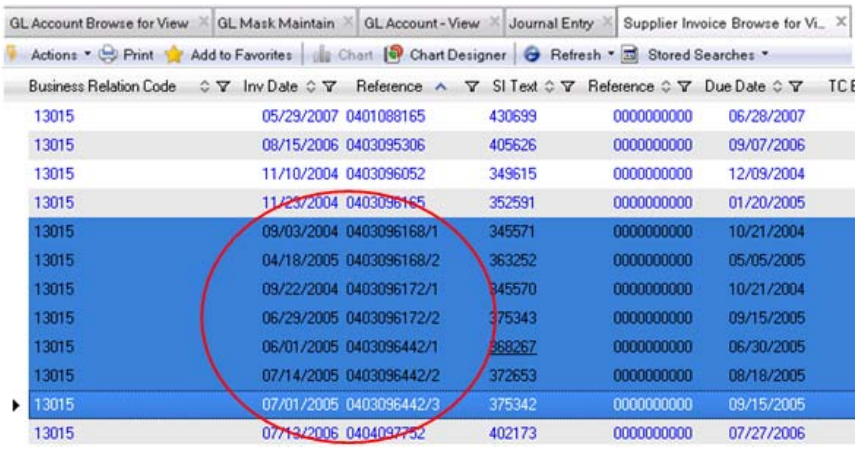
Note If GL transaction consolidation was performed in the pre-conversion database, and multiple AR control accounts are in use, it is likely that compensating balances remain on the AR conversion accounts. You can correct these balances post-conversion using a journal entry.

AP Transactions

The conversion of AP transactions is similar to the conversion process for AR transactions. One difference, however, is the treatment of duplicate invoice numbers.

QAD Enterprise Edition prohibits duplicate invoice numbers for a single supplier. Standard Financials provides a warning in this situation. QAD Enterprise Financials displays an error when it finds duplicate invoice numbers. When the conversion encounters a duplicate invoice number for a supplier, it appends a suffix of /n to the invoice number. The /n is any number, such as /1, /2, and so on. The following figure illustrates the use of this suffix.

Fig. C.1
Duplicate Invoice Number Indication



When the conversion is complete, the remaining balance of the conversion accounts must equal the pre-conversion difference between the GL and sub-ledger. If additional balances remain, investigate and resolve them.

Residual balances in the AR or AP conversion accounts can result from transactions previously posted to the control accounts that did not truly relate to AR or AP. These differences are normally indicated in the Pre-conversion Integrity Report. Complete the following steps to identify specific errors:

- Balance aging by account to the GL for each month.
- Add the balances in all conversion accounts for the sub-ledger (AR or AP) to the converted control accounts for the sub-ledger. This total should be the same as the control account total before conversion.

You can post journal entries to conversion accounts, but not to control accounts, except for Fixed Assets Control accounts.

If the Inventory Control or WIP Control accounts are out of balance, use Issues – Unplanned (3.7, `icunis.p`) or Receipts – Unplanned (3.9, `icunrc.p`) to correct the account balance.

Project Status Codes

The conversion creates two default project status codes:

- Open, with a system status of Open
- Closed, with a system status of Closed

Project Groups

The conversion creates a single default project group named All.

Security

Users

The conversion creates an active user for each user record in the database being converted, where the user ID is not blank. The conversion also creates user domains for each domain in the User Domain Detail (udd_det) table, and creates a user entity record linking the user to each entity in that domain. The conversion marks the user record as enabled when the source record is active.

User Roles

A user role is created for each User Role Detail (usrgd_det). The role name is derived from the linked user record. A User Role Domain (UsrRoleDomain) and User Role Entity (UsrRoleCompany) are also created based on the domain of the User Role Detail.

Daybooks

For each existing daybook with a non-blank code, the conversion creates a daybook record in the shared set of the domain to which the daybook belongs.

The converted daybook is assigned a daybook type of Journal Entry, a layer code of Official, and a control type of Operational. If the source daybook has not reached its expiration date, the daybook is marked as active in the converted database. Otherwise, the daybook is marked as inactive. The conversion ensures that the associated NRM sequence for active daybooks uses the new format of YYYY<*daybook code*>000000001.

Additional active default daybooks are created in each shared set with the type of Journal. The daybooks are the daybooks prompted for in the pre-conversion Conversion Parameters Utility.

Generalized Codes

Deleted Generalized Codes

A number of tables in QAD Enterprise Financials hold data previously stored in Generalized Code Master (code_mstr). The conversion moves the following data from generalized codes to their new tables:

- Each generalized code defined for the tx2_tax_usage field is moved to the Tax Usage (txu_tax_usage) field in the Tax Usage Master (txu_mstr) table.
- Each generalized code defined for the taxc_taxc field is moved to the Tax Class (txcl_tax_cls) field in the Tax Class Master (txcl_mstr) table.
- Each generalized code defined for the txt_tax_type field is moved to the Code (txty_tax_type) field in the Tax Type Master (txty_mstr) table.
- Each generalized code defined for the txe_tax_env field is moved to the Tax Environment (txe_tax_env) field in the Tax Environment Master (txe_mstr) table.

New Generalized Codes

The conversion validates that the data required in Generalized Codes exists. Generalized Code Master (code_mstr) table entries are created for the following items:

- Each state code on Employee Records (emp_mstr.emp_state) and other address records (ad_mstr.ad_state) must exist in generalized codes with a code_fldname value of ad_state.
- Each customer type (cm_mstr.cm_type) must exist in Generalized Codes with a code_fldname value of cm_type.
- Each county code on Addresses Records (ad_mstr.ad_county) must exist in Generalized Codes with a code_fldname value of ad_county.
- Each tax class on address records (ad_mstr.ad_txc) must exist in Generalized Codes with a code_fldname value of ad_txc.
- Each voucher type (vo_mstr.vo_type) must exist in Generalized Codes with a code_fldname value of vo_type.
- Each supplier type (vd_mstr.vd_type) must exist in Generalized Codes with a code_fldname value of vd_type.

SAF Codes

The following Supplementary Analysis Field (SAF) codes are created using generalized codes:

- Item Group (pt_group)
- Item Type (pt_part_type)
- Customer Type (cm_type)
- Supplier Type (vd_type)
- Region (cm_region)

Additional SAF codes are created for sites and non-blank product lines. SAF codes are system-level data and are not domain-specific.

Exchange Rates

The conversion sets the latest exchange rate for each currency as the active exchange rate for that currency, even if the exchange rate had an expiration date before conversion. Exchange rates are effectively open until a new exchange rate is created for that currency with a later start date.

In QAD Enterprise Financials, any exchange rates between two specific currencies are always stated in terms of the From currency. For example, if an exchange rate is created from Currency 1 to Currency 2, you cannot then create an exchange rate from Currency 2 to Currency 1. If this situation exists pre-conversion, the conversion updates the exchange rates so that they are consistent.

Supplier Types

The conversion creates active supplier types for each value found in the Generalized Codes table, where the code_fldname value is vd_type and the generalized code is not blank. Supplier types are system-level data in QAD Enterprise Financials.

Purchase Types

The conversion creates active purchase types for each value found in the Generalized Codes table, where the code_fldname value is vo_type and the generalized code is not blank. Purchase types are system-level data in QAD Enterprise Financials.

Customer Types

The conversion creates active customer types for each value found in the Generalized Codes table, where the code_fldname value is cm_type and the generalized code is not blank. Customer types are system-level data in QAD Enterprise Financials.

Credit Terms

The conversion creates active credit terms with a type of Normal based on the credit terms code in the database being converted. If the pre-conversion credit term has multiple due dates, a staged credit term is created. In this instance, the associated credit term has a type of Staged rather than Normal.

Credit terms are system-level data in QAD Enterprise Financials. If the pre-conversion database contains multiple credit terms with the same name, but in different domains, and with different terms, the conversion creates only one credit term for the credit terms code.

Credit Ratings

The conversion creates active credit ratings for each value found in the Generalized Codes table, where the code_fldname value is cm_cr_rating and the generalized code is not blank. Credit ratings are system-level data in QAD Enterprise Financials.

Invoice Status Codes

By default, the conversion creates the following Invoice Status Codes:

AP-INITIAL for use with Initial Status supplier items (Unconfirmed Vouchers)

- Description = Default – GL Allocation Status
- Lock Payment = True
- Invoice Approved = False
- Allocation Status = No Allocation
- Initial Status = True
- Receiver Matching = True
- Status After Match = AP-RMALLOC

AP-NOALLOC for use with unallocated supplier invoices (Waiting Expense Vouchers):

- Description = Default – No Allocation Status
- Lock Payment = False
- Invoice Approved = False
- Allocation Status = No Allocation
- Initial Status = False
- Receiver Matching = True
- Status After Match = AP-RMALLOC

AP-RMALLOC used as the “Status After Matching” for the AP-NOALLOC Invoice status code

- Description = Default – Receiver Matched Status
- Lock Payment = False
- Invoice Approved = True
- Allocation Status = Allocation
- Initial Status = False
- Receiver Matching = True
- Status After Match = (Not Applicable)

AP-GLALLOC used for the conversion of supplier items

- Description = Default – GL Allocation Status
- Lock Payment = False
- Invoice Approved = True
- Allocation Status = Allocation
- Initial Status = False
- Receiver Matching = False
- Status After Match = (Not Applicable)

AR-ALLOC used for the conversion of customer items

- Description = Default – GL Allocated Status
- Lock Payment = False

- Invoice Approved = True
- Allocation Status = Allocation
- Initial Status = False
- Receiver Matching = False
- Status After Match = (Not Applicable)

AR-CONTESTED for use with contested open customer items

- Description = Default – AR Contested Status
- Lock Payment = False
- Invoice Approved = False
- Allocation Status = No Allocation
- Initial Status = False
- Receiver Matching = False
- Status After Match = (Not Applicable)

You can easily change these codes and their descriptions post-conversion. Any change is reflected in AR and AP invoices where these codes are referenced.

Payment Status Codes

Customer Payment Status Codes

The conversion creates all, or a subset of, the customer payment status codes listed in the following table.

Table C.3
Customer Payment Status Codes

Payment Instrument	Status
Check	Initial
Check	Paid
Check	Bounced
Draft	Initial
Draft	For Collection
Draft	Paid
Draft	Paid Conditionally
Draft	Bounced
Draft	Allocated

Determining Factors:

- The conversion always creates customer payment status codes for check payment instruments. The status codes are created for each GL bank account.
- The conversion creates customer payment status codes for the payment instrument Draft for Accounts Receivable Master (ar_mstr) records with a type (ar_type) of D. The status codes are only created for the GL bank accounts against which the draft transactions were processed.

Note Standard Financials does not use AR payment in process (PIP) accounts. Therefore, the conversion does not create payments with a status of For Collection for customer payments of type Check. All payments are converted with the status Paid and an Open indicator of No.

Supplier Payment Status Codes

The conversion creates all, or a subset of, the supplier payment status codes listed in the following table.

Table C.4
Supplier Payment Status Codes

Payment Instrument	Status
Check	Initial
Check	For Collection
Check	Paid
Check	Void
Transfer	Initial
Transfer	For Collection
Transfer	Paid
Transfer	Void

Payment Instrument	Status
Electronic Transfer	Initial
Electronic Transfer	For Collection
Electronic Transfer	Paid
Electronic Transfer	Void
Draft	Initial
Draft	For Collection
Draft	Paid
Draft	Paid Conditionally
Draft	Void

Determining Factors:

- The conversion always creates payment statuses for payment instruments of type Check. The statuses are created for each GL bank account.
- The conversion creates payment statuses for payment instruments of type Transfer when an Accounts Payable Master (ap_mstr) record exists with a type (ap_type) of CK and a Check Form (ap_ckfrm) of 3.
- The conversion creates payment statuses for payment instruments of type Electronic Transfer when an ap_mstr record exists with a type (ap_type) of CK and a check form (ap_ckfrm) of 4.
- The conversion creates payment statuses for payment instruments of type Draft when an Accounts Payable Master (ap_mstr) record exists with a type (ap_type) of CK and a check form (ap_ckfrm) of 5, 6 or 7.
- For payment instruments Check, Transfer, and Electronic Transfer, the conversion only creates the payment status For Collection if the Use Payment in Process Acct (apc_ctrl.apc_pip) field in AP Control (28.24, `appm.p`) is set to Yes. If this status is created, the Payment in Process (PIP) account defined in Bank Maintenance (bk_mstr.bk_pip_acct) is assigned to the GL account field for the Paid statuses.

Note the following:

- If the Use Payment in Process Acct (apc_ctrl.apc_pip) field in AP Control is set to No, AP payments for the payment instruments Check, Transfer, and Electronic Transfer are converted with the status Paid and with an Open indicator of No.
- If the Use Payment in Process Acct (apc_ctrl.apc_pip) field in AP Control is set to Yes, AP payments for the payment instruments Check, Transfer, and Electronic Transfer are converted with the status For Collection and with an Open indicator of Yes.

When PIP accounts are not used in the pre-conversion environment, regardless of the payment status, the conversion assigns all supplier payments a status of Paid.

QAD Enterprise Financials requires that PIP accounts be entered for payments with a status of Paid. However, unless the Conditional Collection status is used, the PIP account is retrieved from the For Collection status. The GL account is not required for this status and the use of PIP accounts is optional.

Regardless of whether PIP accounts are used or not in the pre-conversion environment, transactional processing can continue following the conversion to QAD Enterprise Edition. The processing is done without having to change static and transactional data.

If you want to implement PIP accounts, you can do this pre- or post-conversion.

Pre-conversion

You can create the PIP account, populate the required bank accounts with the new PIP account, and change the Use PIP Accounts field to Yes in AP Control. For the post-conversion position to be correct, ensure that the GL balance for the PIP account matches the balance of the open checks. You cannot adjust this post-conversion because QAD Enterprise Edition does not allow journal entries to customer or supplier payment accounts.

If you follow this approach, the conversion creates an open payment with the For Collections status.

Post-conversion

You can create a payment status of For Collection and assign the required PIP account. This approach is used going forward. Maintain a manual list of open payments in the pre-conversion database until all of the payments are cleared or closed.

Supplier Bank Data

QAD Enterprise Financials requires that at least one active bank be defined for each supplier. If supplier bank data is maintained in Supplier Maintenance in the pre-conversion database, the data is used during the conversion. You can optionally create a default supplier bank account.

Supplier bank accounts are converted as active or inactive, depending upon the effective dates defined in Supplier Maintenance in the pre-conversion database. A single bank account for each supplier is set as the default bank (the source effective dates determine the default). If the supplier has multiple bank accounts and more than one bank account is effective, the conversion sets all of the supplier bank accounts to active and sets the first bank account in alphanumeric order as the default. Review these settings after the conversion.

Payment Formats

Payment formats are used in customer and supplier payments to define the layout of the payment output. The term payment format is new to QAD Enterprise Financials, and replaces Check Forms in Standard Edition and Payment Methods in European Accounting.

The conversion determines the payment format using a combination of the bank and payment method on the supplier, customer, customer invoice, supplier invoice, or customer or supplier payment.

Non-European Accounting

The conversion creates a payment format for each of the AP check forms (1-8) in the format Check Form 1, Check Form 2, and so on.

The conversion also creates a default AR check payment format because earlier QAD versions did not have this concept. The AR check payment format is called AR Check.

If the pre-conversion system uses drafts (meaning that Accounts Receivable Master records exist where the type is D), the conversion also creates a default AR draft payment format called AR Draft.

European Accounting

QAD Enterprise Edition Financials includes a list of supported payment formats. These formats are defined in XML files that you can obtain from QAD Support. The conversion loads the supported formats from the directory specified in the Conversion Parameters Utility.

The conversion attempts to convert non-standard (that is, non-QAD) payment formats to Enterprise Edition. However, the conversion is not always successful. The expected outcome is that any customers, suppliers, and banks that reference non-standard payment formats are configured correctly. However, the actual payment format details (the file output format) could be less than 100% correct due to differences before conversion to Enterprise Edition. Review and modify the output from this process in Enterprise Edition as required.

You can identify non-standard payment formats in Enterprise Edition by their names, which are the same as in the pre-conversion environment (for example, `euqptswp.p`). Standard, supported Enterprise Edition payment formats have names such as GENERIC-PAY-AP or DE-DTAUS-AR.

Consolidation

The conversion creates consolidation cycle records based on GL consolidation sets in the source database. Each of these records requires a unique daybook for each consolidation entity. The conversion creates these records using the format:

<Four character daybook code provided in Conversion Parameters Utility><Entity Code>

After the conversion, review and update the following data as necessary:

- Default sub-account, project, cost center, rounding GL account, and tax codes
- Default SAF values for GL accounts, cost centers, and projects
- Consolidation cycle status
- Daybooks associated with the management and transient layers
- COA cross-reference codes

There are considerable differences between the setup required by the From-Acct Cross-Reference Maintenance function in previous QAD applications versions and the account and sub-account cross-references in QAD Enterprise Financials. Therefore, consolidation cross-reference data is not converted. Instead, use Excel integration and copy the existing records to configure this data after conversion.

Voucher Detail Records

The conversion does not reference Voucher Detail (vod_det) records.

Pre-conversion

The Pre-conversion Integrity Report reports any missing Voucher Detail records, but this output is for information only, and does not stop the conversion.

Conversion

The conversion only references Voucher Detail records when converting AP invoices (converting ap_mstr and vo_mstr records to CInvoice records). The conversion uses the Voucher Detail data in several ways:

- The conversion uses the vod_det.vod_dy_num field to populate the Posting.PostingOriginDaybookNumber field for the posting records created.
- If the converted record is a waiting expense voucher transaction, the conversion uses the voucher detail data to determine the waiting expenses amounts.
- When converting unconfirmed vouchers, the conversion removes Voucher Detail records to allow receiver matching after conversion.

Unconfirmed Supplier Vouchers

When unconfirmed vouchers are converted to Enterprise Edition, GL distribution or receiver matching data is lost and must be re-entered.

In earlier QAD versions, the Accounts Payable module allows you to enter vouchers marked as unconfirmed. In Standard Financials, you can match unconfirmed vouchers to purchase order receipts and you can enter GL distribution lines without creating actual GL transactions.

In earlier QAD versions, you could modify or delete unconfirmed vouchers entirely. The vouchers are not widely visible within the system, and are not available for payment. Once the voucher and GL distribution or matching are approved, you can use a separate confirmation function to set the voucher status to confirmed. At that point, the system creates the required GL transactions, prohibits further modification to the voucher, and makes the voucher available for payment.

In Enterprise Edition, the creation and timing of general ledger postings within the supplier invoice process is controlled using invoice status codes. However, Enterprise Edition does not have an invoice status code for creating supplier invoices with detailed GL posting and matching data that can later be reversed or deleted. The Initial invoice status code provides a similar, but not equivalent, functionality. Supplier invoice header data (for example, the supplier, PO number, and invoice total) is held, but GL or matching data is not. Therefore, converting unconfirmed vouchers erases the GL distribution or receiver matching data held against the supplier invoice.

Appendix D

Conversion Troubleshooting

This appendix provides conversion troubleshooting information.

Introduction

This appendix describes issues that prevent a successful conversion. These issues include the following:

- Errors in Data Preparation Report
- Conversion Errors in yab.log
- Errors in conversion.log
- Progress Errors During Conversion Execution

Errors in Data Preparation Report

Symptom: `Data Preparation Report is not clean` error message appears after you input the source database.

Before you can execute a conversion, you must run the Data Preparation Report on the source database. The report must indicate zero errors.

To fix the issues, run the Data Preparation Report against the source database, and correct any reported errors.

Conversion Errors in yab.log

Index Rebuild Errors

The `yab.log` file contains the output from rebuilding the indexes in the QAD Enterprise Edition database during the conversion execution stage.

The new unique index oid_<table name> can cause these errors. OID fields were available in some pre-QAD Enterprise Edition versions, but uniqueness was not enforced. This new index is used to enforce the uniqueness of OID fields.

Analyze and correct these errors before proceeding with the conversion. For OID fields, you can reset them to zero and the conversion regenerates them. Some OID fields are used as foreign fields and require manual correction.

Errors in conversion.log

Invalid Characters in Database Field

Error message: `field cannot contain a comma, a pipe or any unprintable character`

QAD Enterprise Edition does not support commas, pipes, or unprintable characters in some fields. Remove these characters from the source database or replace them with supported characters such as a semicolon.

Number of Characters in Database Field Exceeds Limit

Error message: `value is too long`

QAD Enterprise Edition validates field length based on data definitions. If the value entered in the field exceeds the length defined in the database, QAD Enterprise Edition reports an error during conversion.

You can correct these errors by shortening or truncating some characters to a length that is less than, or equal to, the length defined in the database.

Role Name Contains Unsupported Characters

Error message: `The role name may not contain the following characters: '*', ',' or '!'`

QAD Enterprise Edition's role name does not support the characters "*, "," or "!".

Replace them with the supported characters in usrg_group_name of the source database.

Pay Format Directory Not Found

Error message: `***** Error: Cannot find Pay Format Directory`

The `payformats` XML directory was entered incorrectly using the Conversion Parameters utility or the directory does not contain any of the required XML files.

Confirm that the directory entered is valid and contains the correct XML files.

Progress Errors During Conversion Execution

Default Daybook Codes Not Found

Error message: `Record Already exists with Daybook Code = 0.`

The conversion could not find the default daybook codes because they were not entered or an earlier error prevented the conversion from writing these values to the XML file.

Update the values using the Conversion Parameters utility.

Could Not Start Financial Session

Error message: `Could not start financial session. -5`

The possible causes of this error and the suggested actions are:

- The financial application server is not running. Start the financial application server.
- The license keys are out of date. At some point, the conversion was stopped and restarted.

Appendix E

Log Files

This appendix describes the conversion log files.

Introduction

Conversions are executed within the YAB console. The `conversion.log` file is the primary source of conversion logging information.

In addition to the `conversion.log` file, several other log files can be created during a financial conversion, especially if certain error conditions occur.

The `conversion.log` file and the other log files, described in the following table, are created in the `<application_installation>/build/logs directory`.

Table E.1
Financial Conversion Log Files

Log File Name	Description
`AddressMaster.log`	When addresses and contacts are created during the conversion, Service/Support End User Detail (eud_det) records with a Name (eud_sort) field exceeding 24 characters are truncated. This file contains a log of truncated records.
`APInvoiceInvalidData.log`	This file contains a log of AP Invoices which contained some invalid data, but were processed during the conversion. Examples of such invalid data are AP invoices in which the company, currency, or daybook information was invalid or referenced a nonexistent record. The AP Invoice record is still fully converted, but post-conversion action could be required.
`APPaymentInvalidData.log`	This file contains a log of AP Payments which contained invalid data, but were processed during the conversion. Examples of such invalid data would be AP payments in which the company, currency, or supplier information was invalid or referenced a nonexistent record. The AP Payment record is still fully converted, but post-conversion action could be required.
`ARInvoiceInvalidData.log`	This file contains a log of any AR Invoices which contained some invalid data, but were processed during the conversion. Examples of such invalid data would be AR invoices in which the company, currency, or daybook information was invalid or referenced a nonexistent record. The AR Invoice record is still fully converted, but post-conversion action could be required.
`ARPaymentInvalidData.log`	This file contains a log of AR payments which contained some invalid data, but were processed during the conversion. Examples of such invalid data are any AR payments in which the company, customer, or daybook information is invalid or references a nonexistent record. The AR Payment record is still fully converted, but post-conversion action could be required.
`BankFormatInvalidData.log`	This file contains a log of errors encountered when determining the bank format details for a customer or supplier bank. An example of such data is an invalid BankNumber and payment format combination. The record is still fully converted, but post-conversion action could be required.
`PaymentStatusInvalidData.log`	This file contains a log of any PIP (Payment in Process) accounts that were defined with an incorrect GL account type.
`ConversionAccountPostings.log`	This file contains a log if there are any postings against the control account of a customer, or supplier, but against the mirror of the transaction account.

Log File Name	Description
ARBankNumberChange.log	This file contains a log of any records in which it was necessary to default the bank details for AR transactions. One of two methods can be used to default the values: 1. The bank details associated with ar_bank are used instead of ar_acct. This default is used if ar_acct is not defined as a valid bank. 2. A default bank number based on the customer and customer bank number. This default is used if both ar_acct and ar_bank fail to provide a valid bank.
BankInvalidData.log	This file contains a log of any errors encountered when determining the supplier bank details. An example of such invalid data would be blank pay format data for a supplier.
GLInvalidData.log	This file contains a log of any invalid data encountered when converting GL History (gltr_hist) records. Examples of such invalid data are an invalid GL Entity (gltr_hist.gltr_entity) or GL Account (gltr_acct).
PostingBalance.log	This file contains details of any additional posting that the conversion had to make to ensure a GL Transaction Posting balances. This log entry provides traceability for these newly created records.
BankConsolidation.log	During the pre-conversion process, you might need to specify replacement Payment in Process (PIP) accounts and replacement Drafts Payable accounts within the GL Account Type Utility. The system prompts you for these accounts when bank records exist that contain the same Domain code (where applicable), Entity code, and Cash account, but where the PIP accounts or Drafts Payable accounts differ. The account entered is used going forward. Therefore, any transactions that were posted against the original PIP or Drafts Payable account are converted using the replacement account. This log file contains a record of such transactions where the PIP or Drafts Payable account were replaced with the new PIP or Drafts Payable account.

Reviewing Log Files After Conversion

Once a conversion finishes, review the `conversion.log` file for any errors and warnings. Also review any financial conversion logs that were created.

Have a financial consultant review all data-related errors and warnings to determine whether corrective action is necessary.

Appendix F

AIM Conversion

This appendix describes the conversion of the AIM database.

Introduction

By default, the `conversion-pre-ee-upgrade` process does not execute the AIM database conversion process `conversion-qadaim-run`.

Configuration to Enable AIM DB Conversion

The following two configuration settings are required to configure the `conversion-pre-ee-upgrade` process so that the `conversion-qadaim-run` process is executed.

Table F.1
AIM DB Conversion

Properties	Default	Description
conv.pre-ee.src-db-aimdb.dir	none	The directory where source AIM database is located.
conv-ui.AIMDOMAINS	none	The domain to be converted.

Optional Configuration

The `conversion-qadaim-run` process also has a number of optional configuration properties that are detailed in Table F.2. These properties have default values and do not need to be explicitly set.

Table F.2
Optional Configuration

Properties	Default	Description
conv.pre-ee.src-db-aimdb.physicalname	aimprod	The unqualified physical name of the source AIM database.
conv.pre-ee.src-db-aimdb.logicalname	aimsrc	The logical name of the source AIM database.
conv.pre-ee.src-db-aimdb.codepage	${db.qaddb.codepage}	The codepage of the source AIM database.

Appendix G

Snapshots

This appendix describes the conversion snapshot feature.

Introduction

You can configure the conversion process to pause at a predefined point to allow you to make backups (snapshots). Snapshots allow you to easily restore and restart the conversion at the point where the snapshot was made.

Pause the Conversion

You can pause the conversion in order to take a snapshot by executing the following command:

```
> yab -debug-pause -break:conversion-tlc-create conversion-pre-ee-upgrade
```

This stops the conversion process at the `conversion-tlc-create` stage.

```
...
60/81 database-qaddb-index-rebuild                          OK (1:23 m)
61/81 conversion-tlc-create
Press ENTER to run 'conversion-tlc-create' ('s' skip, 'a' run all, 'q' quit).
>
```

Enter `q` to quit.

Make a Snapshot

To back up the database, execute the following command:

```
> yab database-qaddb-backup database-qadadm-backup -tag:conv-snapshot
```

Verify the snapshot has been created by executing:

```
> yab database-backup-list
```

Restart the Conversion

To restart the conversion after successfully creating a snapshot from the point it was paused, execute the following:

```
> yab -debug-pause -break:conversion-tlc-create -skip conversion-pre-ee-upgrade -clean
```

Enter a to run all remaining conversion processes.

Restore a Snapshot

To restore your system to the conversion snapshot, you must first stop the environment.

```
> yab stop
```

Then restore the conversion snapshot by executing the following:

```
> yab database-qaddb-restore database-qadadm-restore -tag:conv-snapshot
```

You can now restart the conversion as described in Restart the Conversion.

Appendix H

Validate the Conversion Process (conversion-ee-upgrade)

This appendix describes validating your upgrade.

Introduction

After the conversion completes, you should validate that it was successful with the following procedure.

Validate the Upgrade Execution

The upgrade progress can be monitored by examining the following log files:

```
> ../<application-directory>/logs/yab.log
> ../<application directory>/logs/conversion.log
```

Once the conversion has completed, the following checks should be done.

Review Console Output – Build Successful

Ensure that the `conversion-ee-upgrade` process completed successfully. The output should be similar to the following.

Note The timings vary, depending on the size of the database and the server specification.

```
host: /dr02/qadapps/ee $ yab conversion-ee-upgrade

                      conversion-ee-upgrade (29 tasks)
------------------------------------------------------------------------------
1/29  conversion-stage                                  OK (0.101 s)
2/29  conversion-mfg-package-stage                      OK (5.883 s)
3/29  conversion-fin-package-stage                      OK (6.519 s)
4/29  conversion-mfg-get-db-version                     OK (0.480 s)
5/29  conversion-fin-get-db-version                     OK (0.322 s)
6/29  conversion-mfg-create-execution-list              OK (0.520 s)
7/29  conversion-fin-create-execution-list              OK (0.325 s)
8/29  conversion-mfg-special-dump-run                   OK (0.777 s)
9/29  conversion-fin-special-dump-run                   SKIPPED (0.004 s)
10/29 conversion-mfg-qaddb-data-delete                  OK (5.184 s)
11/29 conversion-fin-qaddb-data-delete                  SKIPPED (0.005 s)
12/29 conversion-mfg-qadadm-data-delete                 OK (37.698 s)
13/29 conversion-fin-qadadm-data-delete                 SKIPPED (0.016 s)
14/29 database-qaddb-structure-update                   OK (0.215 s)
15/29 database-qaddb-schema-update                      OK (7:59 m)
16/29 database-qaddb-schema-index-rebuild               OK (43.169 s)
17/29 oid-code-update                                   SKIPPED (0.022 s)
18/29 database-qaddb-schema-snapshot                    OK (9.732 s)
19/29 database-qadadm-structure-update                  OK (0.060 s)
20/29 database-qadadm-schema-update                     OK (24.327 s)
21/29 database-qadadm-schema-index-rebuild              OK (0.303 s)
22/29 database-qadadm-schema-snapshot                   OK (0.761 s)
23/29 database-qadhlp-structure-update                  OK (0.094 s)
24/29 database-qadhlp-schema-update                     OK (7.572 s)
25/29 database-qadhlp-schema-index-rebuild              OK (0.350 s)
26/29 database-qadhlp-schema-snapshot                   OK (0.266 s)
27/29 conversion-mfg-qaddb-run                          OK (16.345 s)
28/29 conversion-fin-qaddb-run                          OK (15.368 s)
29/29 conversion-mfg-qadadm-run                         OK (6.928 s)
------------------------------------------------------------------------------

   BUILD SUCCESSFUL (11:13 m)
```

Note Examining the output for each of these processes is a good starting point in confirming or troubleshooting the conversion-ee-upgrade process.

Review yab.log Output

You should thoroughly review each process associated with the `conversion-ee-upgrade` process in the `yab.log` file. To help navigate the `yab.log` file, you can use the following text to find each process execution entry “BuildContext - ”.

To locate the log content associated with determining the source database version, search the `yab.log` file for:

```
"BuildContext - conversion-mfg-get-db-version"
```

To locate the log content associated with the qaddb conversion step, search the `yab.log` file for:

```
"BuildContext - conversion-mfg-qaddb-run"
```

Review conversion.log Output

You should thoroughly review each conversion process in the corresponding `conversion.log` file. To help navigate the `conversion.log` file, you can use the text in Table 3.4 to find each process execution entry that involves “Executing step:”.

For example, to locate the log content associated with determining the source database version, search the `conversion.log` file for:

```
"Executing step: get-db-version.p"
```

Review QADDB Conversion Output

You should thoroughly review the qaddb conversion in the `conversion.log` file. To locate the log content associated with running the qaddb conversions, search the `conversion.log` file for:

```
"Executing step: run-qaddb-conv.p"
```

Rerun an EE Upgrade

If the upgrade did not complete successfully and terminated prematurely, you must first determine the root cause of the problem and correct it in the source environment.

Once the problem has been resolved, follow these steps to restart the upgrade:

1. Restore the source databases.
2. Run the YAB commands to reset the conversion state and refresh the environment configuration.

   ```
   > yab conversion-ee-upgrade-reset
   > yab –r info
   ```

3. Rerun the upgrade.

   ```
   > yab conversion-ee-upgrade
   ```

Product Information Resources

QAD offers a number of online resources to help you get more information about using QAD products.

QAD Forums (community.qad.com)
Ask questions and share information with other members of the user community, including QAD experts.

QAD Knowledgebase (knowledgebase.qad.com)*
Search for answers, tips, or solutions related to any QAD product or topic.

QAD Document Library (documentlibrary.qad.com)
Get browser-based access to user guides, release notes, training guides, and so on; use powerful search features to find the document you want, then read online, or download and print PDF.

QAD Learning Center (learning.qad.com)*
Visit QAD's one-stop destination for all courses and training materials.

*Log-in required

Index

V