Microsoft® Business Solutions – Solomon

FastForward Travel Guide

Release 6.0

Microsoft® Business Solutions

05/2004

The names of companies, products, people, and/or data used in window illustrations and sample output are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted.

Warranty Microsoft Business Solutions and Microsoft Corporation disclaim any warranty regarding the sample code contained in this documentation, including the warranties of merchantability and fitness for a particular purpose.

Limitation of liability The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Great Plains Software, Inc. or Microsoft Corporation. Great Plains Software, Inc. and Microsoft Corporation assume no responsibility or liability for any errors or inaccuracies that may appear in this manual. Neither Great Plains Software, Inc., Microsoft Corporation nor anyone else who has been involved in the creation, production or delivery of this documentation shall be liable for any indirect, incidental, special, exemplary or consequential damages, including but not limited to any loss of anticipated profit or benefits, resulting from the use of this documentation or sample code.

Publication date May 2004

Table of Contents

Introduction

About This Guide

This guide discusses the setup and use of the Solomon IV FastForward™ 6.0 tools: FastForward *Integrity Wizard, FastForward DB Creation* and FastForward *Transformation Wizard*.

Purpose of the FastForward 6.0 Travel Guide

The *FastForward 6.0 Travel Guide* presents information needed to upgrade Solomon IV for Windows version 2.06 databases to Solomon IV version 6.0 databases. The material includes a comprehensive discussion of the migration process, along with step-by-step instructions for using the FastForward 6.0 tools. Microsoft Business Solutions suggests completely reviewing this guide and preparing an upgrade plan before you begin the Solomon IV version 6.0 database upgrade process.

What is Covered in the Travel Guide?

The Travel Guide consists primarily of procedures and checklists that describe how to perform the various tasks associated with FastForward 6.0 *Integrity Wizard* and FastForward 6.0 *Transformation Wizard*. The Travel Guide also contains topics that help you become better acquainted with the underlying functionality of the migration tools and guidelines for completing the overall migration process.

Who Should Use the Travel Guide?

The Travel Guide is designed as a reference document for completing database migrations. Since each database presents a unique set of circumstances, this document should be used as a tool for assisting in each migration performed. The checklists included in the guide are also available with the FastForward 6.0 tools under the “Documentation” subdirectory.

How to Use the Travel Guide

Read the entire guide prior to installing or using the FastForward 6.0 tools. The Travel Guide presents all procedures and steps required for installing the tools as well as completing the upgrade. To assist you in locating information, the Travel Guide contains:

- A Table of Contents of logically organized activities and tasks.
- An alphabetized Index of the information provided in the Travel Guide.

Introducing the FastForward 6.0 Tools

FastForward 6.0 includes three software tools: FastForward *Database Creation*, FastForward *Integrity Wizard* and FastForward *Transformation Wizard*.

The FastForward *Database Creation* is used to create Solomon IV destination databases.

FastForward *Integrity Wizard* is a standalone application that may be run on the Solomon IV 2.06 source database as many times as necessary to identify and correct irregular conditions.

In addition, *Integrity Wizard* produces the *Integrity Wizard Analysis Report* that provides information on other key considerations for the overall migration such as key record counts for each module, existing customizations, the potential for migration to Select Edition, existing templates, etc. Each section of this report is designed to help you determine the magnitude of the work involved in migrating your database.

Integrity Wizard is written in Microsoft® Visual Basic® 6.0 and uses Pervasive.SQL ODBC connections to link to the source database. It works by executing a suite of specially designed SQL (structured query language) scripts that check the integrity of the database. *Integrity Wizard* checks for a number of potential error conditions such as unreleased or unposted batches, invalid date values, violations of a table's unique key and mismatched records. In addition, it also has the capacity to record and repair a number of common problems.

However, in some cases, database error conditions need to be resolved manually. In these cases, *Integrity Wizard* has the power to export reference information to Microsoft Excel for further evaluation. These Excel worksheets provide information necessary to investigate and repair problem situations. After the irregular conditions in the database have been resolved, *Integrity Wizard* should be run again to ensure that all irregular conditions were successfully resolved. A clean bill of health from *Integrity Wizard* means that the next step of the migration process — *Transformation Wizard* — can proceed.

The FastForward *Transformation Wizard* is the utility used to migrate the source database to Solomon IV version 6.0. This tool is designed to automate the launch of specifically designed Microsoft Data Transformation Service (DTS) packages that pump accounting data from your old Solomon IV database to the Solomon IV destination database structures.

Transformation Wizard uses Pervasive.SQL ODBC connections to link to the existing source Solomon IV 2.06 database. In addition, Object Linking and Embedding (OLE) database connections are used to connect to the destination Solomon IV database. Once successfully connected, *Transformation Wizard* becomes a “data pump” and moves data rapidly between the old and new platforms. *Transformation Wizard* displays process information during execution.

FastForward Version 6.0 Limitations

Version 6.0 of the FastForward tools does not include migration functionality for the following situations:

- Custom tables
- Templates
- Solomon III versions earlier than 2.0c to Solomon IV version 6.0
- Flexkey modifications
- Conversion from competitive applications to Solomon IV version 6.0

Migration assistance for these issues may be contracted through Microsoft Business Solutions Technology Services, mbstserv@microsoft.com.

Migration Process Checklist

The purpose of this checklist is to provide a high level overview of the upgrade process. Detailed checklists for various steps within the migration process are included in the appropriate sections of the Travel Guide. In addition, special considerations must be made for migrations to Select Edition and Solomon IV 2.06 migrations being run on standalone workstations. See “Migrating to Select Edition” on page 37 and “Pervasive.SQL 7 Workstation Edition” on page 27 for detailed instructions addressing these situations.

<table>
<tr><td colspan="4"><b>Migration Process Checklist</b></td></tr>
<tr><td colspan="3"><b>Client:</b></td><td><b>Solomon IV Version:</b></td></tr>
<tr><td colspan="3"><b>Completed by:</b></td><td><b>Date:</b></td></tr>
<tr><td><b>X</b></td><td></td><td colspan="2"><b>Action Item</b></td></tr>
</table>

Migration Process Checklist		
Client:		**Solomon IV Version:**
Completed by:		**Date:**
X		**Action Item**
	1.	Obtain commitment for upgrade from client, establish estimated timeline for upgrade and procure Solomon IV v6.0.
	2.	Obtain copy of client’s database in order to perform a test migration. Although this is not required, it is strongly recommended in order to assess the database condition and time needed to process the live migration. Performing the test migration offsite is strongly recommended.
	3.	Prepare the environment being used for the test migration in accordance with the specifications included in the Installation and ODBC Configuration sections of the Travel Guide.
	4.	Prepare the database for migration based on the requirements listed in the *Source Database Preparation Checklist*.
	5.	Produce the reports needed from the source database as listed in the *Source Database Report Checklist* to facilitate post-migration validation.
	6.	Process the database through *Integrity Wizard* to identify any irregular database conditions that require resolution prior to migration.

Migration Process Checklist		
Client:		**Solomon IV Version:**
Completed by:		**Date:**
X		**Action Item**
	7.	Research any errors returned by *Integrity Wizard*. The Excel spreadsheets it produces provide an audit trail of the records that require repair. Some situations will require consultation with the client prior to repair. Resolutions should be tested during the test migration to ensure effectiveness and avoid delays during the live migration.
	8.	Upon completion of *Integrity Wizard*, print the *Integrity Wizard Analysis Report*. In addition to providing information regarding *Integrity Wizard* results, this report includes important information regarding the source database that must be taken into consideration for the overall upgrade process. For example, the report lists customizations, templates and custom reports that will need to be reviewed and potentially re-created in the Solomon IV v6.0 database. See “Integrity Wizard Analysis Report” on page 128 for explanations of the information included in the report.
	9.	Create an empty database through *Solomon IV FastForward* **-** *Database Creation* (FF.290.00) on the destination server.
	10.	Use *Transformation Wizard* to migrate the database.
	11.	Complete any post-migration data entry into setup screens as required by certain modules.

Migration Process Checklist		
Client:		**Solomon IV Version:**
Completed by:		**Date:**
X		**Action Item**
	12.	Verify that the data migrated as expected by running reports from the migrated database and comparing to the corresponding reports printed earlier from the source database.
	13.	Confirm final live migration schedule with client, taking into consideration any custom reports, templates and customizations that will need to be re-created.
	14.	After the live migration is completed, review and update screen security in *Access Rights Maintenance* (95.270.00).
	15.	Update PVRECs through *Database Administration* (98.270.00).
	16.	Upgrade FRx® from version 4.7 to 6.7.

FastForward 6.0 Installation

Requirements for Installation

An essential component to successful use of the FastForward 6.0 tools is the correct installation of the tools and supporting software components in an environment with the capacity to achieve migration of the database. The majority of migrations will require one workstation and two servers. A typical successful installation would have the FastForward 6.0 tools installed on the workstation, the Solomon IV 2.06 source database installed on one server and the empty Solomon IV destination database on another server.

Following are the ideal configuration requirements for performing migrations:

Requirements for the Workstation

Install the following on the workstation:

- NT Workstation, Windows 98, Windows 2000 or Windows XP
- FastForward 6.0 (which installs the Pervasive.SQL 32-Bit ODBC driver version 2.54 which is needed for Solomon IV 2.06 migrations)
- SQL Server 2000 Client Utilities (Management Tools and Client Connectivity)
- Solomon IV version 6.0 Client
- Pervasive.SQL 7 Workstation Edition (only if migrating Solomon IV 2.06 on standalone workstations)

Requirements for the Source Database Server

Install the following on the server for the source database:

- NT Server, Windows 2000 or Windows XP
- Scalable SQL for NT
- Solomon IV version 2.06 Database Files

Requirements for the Destination Database Server

Install the following on the server for the destination database:

- NT Server, Windows 2000 or Windows XP
- SQL Server 2000
- Solomon IV version 6.0

Configuration Considerations

Minimum (and Recommended) Workstation Configuration Requirements

- Windows NT™ Workstation, Service Pack 4 (Windows 2000)
- 133 mHz Pentium II Processor (300 mHz Pentium II Processor)
- 128 MB of RAM (256 MB of RAM)
- 250 MB hard drive space (When available, the Pervasive.SQL Workstation migrations will require disk space equivalent to four times the size of the source database.)

Note: The stability and performance of the FastForward 6.0 tools are directly proportional to workstation and server configurations.

ODBC Drivers

- Solomon IV version 2.06 requires Pervasive Software ODBC-32, version 2.54 drivers. The drivers are distributed with the FastForward 6.0 tools and should be installed on the workstation being used for the migration.

Note: Refer to the section on ODBC configurations (page **Error! Bookmark not defined.**) for additional information on creating the ODBC connection.

Pervasive.SQL Workstation Edition

Pervasive.SQL 7 Workstation Edition provides the ability to migrate Solomon IV 2.06 databases on a standalone workstation. Pervasive.SQL 7 Workstation Edition may be installed during the standard installation or at a later time by running SETUP.EXE from the PVSW directory within the FastForward 6.0 tools download. Installing Pervasive.SQL 7 Workstation Edition also automatically installs the Pervasive 2.54 ODBC drivers.

Novell NetWare

FastForward 6.0 tools may encounter error situations when used to migrate a Scalable SQL database running on a NetWare server. Current configurations require utilizing NT Server and Scalable SQL for NT for upgrades. This may require obtaining a copy of the client's database and performing the database assessment and migration offsite. If an NT network is not available, the database can be migrated using Pervasive.SQL 7 Workstation Edition on a standalone workstation.

Multiple vs. Single Server Environments

The recommended configurations suggest the use of two servers for the migration process.

Printer Requirements

FastForward 6.0 tools will operate without a printer connection. However, analysis features, such as the ability to print records exported to Microsoft Excel and the *Integrity Wizard Analysis Report* will not be available. These features are valuable components of the overall FastForward 6.0 application as both planning and research tools.

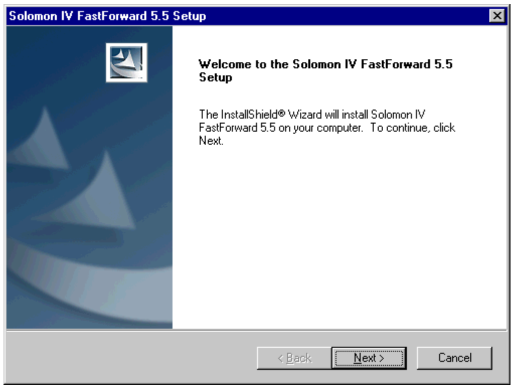
Microsoft Excel 97

Both *Integrity Wizard* and *Transformation Wizard* provide the option to export information to Microsoft Excel 97 or later for research and analysis. Although FastForward 6.0 will function without Microsoft Excel 97 or later installed, the ability to export records with irregular conditions for research and analysis will not be available.

Step-by-Step Installation Instructions

> **To install Solomon IV FastForward:**

1. Double-click on SETUP.EXE in the FastForward 6.0 zip file.
2. After reading the *Welcome* screen, click **Next**.



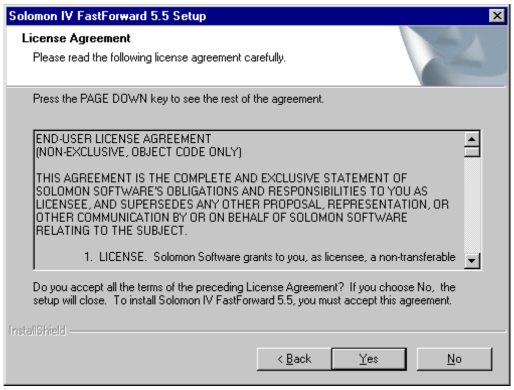
Solomon IV FastForward 5.5 Setup

Welcome to the Solomon IV FastForward 5.5 Setup

The InstallShield® Wizard will install Solomon IV FastForward 5.5 on your computer. To continue, click Next.

Figure 1: Welcome

3. *License Agreement* appears, prompting to accept or decline the Solomon IV license agreement. If you decline, *Exit Setup* appears indicating that the installation is incomplete. Options are available in Exit Setup to resume the installation or to exit.



Please read the following license agreement carefully.

Press the PAGE DOWN key to see the rest of the agreement.

END-USER LICENSE AGREEMENT
(NON-EXCLUSIVE, OBJECT CODE ONLY)

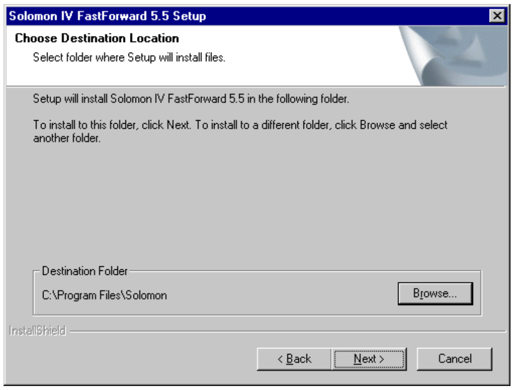
THIS AGREEMENT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF SOLOMON SOFTWARE'S OBLIGATIONS AND RESPONSIBILITIES TO YOU AS LICENSEE, AND SUPERSEDES ANY OTHER PROPOSAL, REPRESENTATION, OR OTHER COMMUNICATION BY OR ON BEHALF OF SOLOMON SOFTWARE RELATING TO THE SUBJECT.

1. LICENSE. Solomon Software grants to you, as licensee, a non-transferable

Do you accept all the terms of the preceding License Agreement? If you choose No, the setup will close. To install Solomon IV FastForward 5.5, you must accept this agreement.

Figure 2: License Agreement

4. *Choose Destination Location* appears.



Choose Destination Location

Select folder where Setup will install files.

Setup will install Solomon IV FastForward 5.5 in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

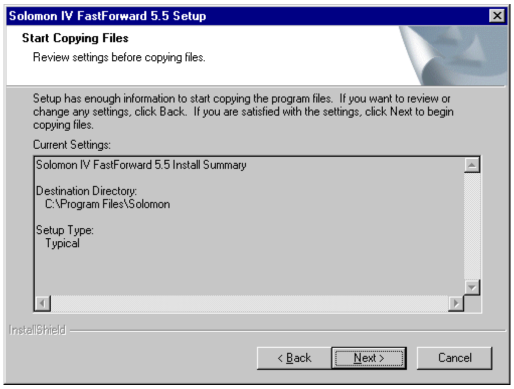
Destination Folder

C:\Program Files\Solomon

Figure 3: Choose Destination Location

- Click **Next** to install to the folder displayed in Destination Folder.
- Click **Browse** to select a different destination folder.

5. *Start Copying Files* appears. Review the setup configurations and click **Next** to begin copying files.



Start Copying Files

Review settings before copying files.

Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

```
Solomon IV FastForward 5.5 Install Summary

Destination Directory:
  C:\Program Files\Solomon

Setup Type:
  Typical
```

Figure 4: Start Copying Files

6. After the file copy is complete, the prompt displays to install Pervasive.SQL 7 Workstation Edition. Respond **Yes** only if FastForward 6.0 is going to be used on a standalone workstation to migrate a Solomon IV 2.06 database. Pervasive.SQL 7 Workstation Edition is only necessary in the following situations:

   - A Solomon IV 2.06 migration is being done on site and client is running the source database on a Novell server, and does not have a license for a Pervasive NT database engine. FastForward 6.0 tools are not compatible with Novell.

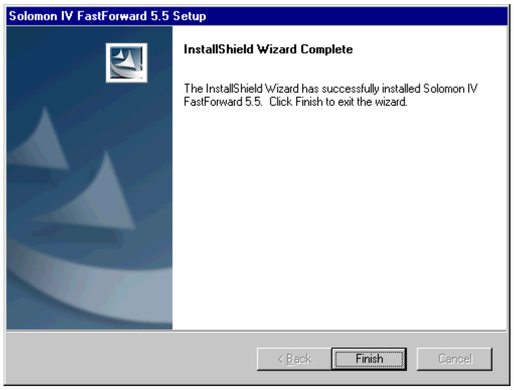
- The source database is for a Solomon IV 2.06 single-user Solomon IV application.
- The migration is being done offsite and an NT server is not available or a Pervasive NT database engine is not available.

Pervasive.SQL 7 Workstation Edition can be installed at a later time by running SETUP.EXE from the PVSW directory within the FastForward 6.0 tools download.

Refer to the “Pervasive.SQL 7 Workstation Edition” on page 27 for additional information.

Important: There are compatibility issues with Solomon IV 2.06 and Pervasive.SQL 7 Workstation Edition on the same workstation due to DLL conflicts. In order to utilize the FastForward 6.0 tools with Pervasive.SQL 7 Workstation Edition, install the tools on a workstation not used to run Solomon IV 2.06. Copy the source database from its existing location to a directory on the workstation where FastForward 6.0 and Pervasive.SQL 7 Workstation Edition are installed.

7. The next prompt is to install 32-Bit Pervasive ODBC drivers. These drivers are required to migrate Solomon IV 2.06 databases and should be installed if they are not currently loaded on the workstation. The drivers can be installed at a later time by running SETUP.EXE from the ODBC directory within the FastForward 6.0 tools download.
8. Once the ODBC drivers are installed, the installation is complete. Click **Finish** in the final screen to exit setup.



InstallShield Wizard Complete

The InstallShield Wizard has successfully installed Solomon IV FastForward 5.5. Click Finish to exit the wizard.

Figure 5: InstallShield Wizard Complete

9. Reboot the workstation to complete setup.

Note: FastForward 6.0 may be uninstalled from *Control Panel*, *Add/Remove Programs*. If there is a need to uninstall FastForward, respond **No to All** when prompted to remove shared components.

ODBC Configurations

ODBC Configuration Overview

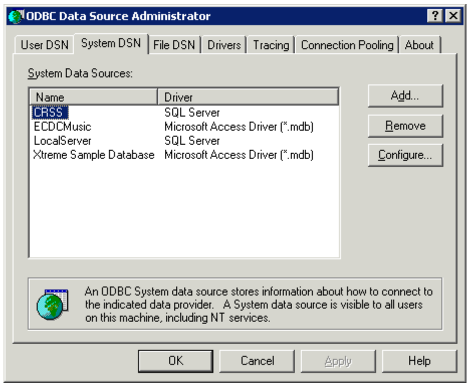
In order for *Integrity Wizard* to function, a successful ODBC connection must be made to the source system and application databases. *Integrity Wizard* uses 32-bit Microsoft SQL Server or Pervasive.SQL 2.54 ODBC connections to link to the source databases. The Pervasive SQL 2.54 ODBC drivers are distributed with the FastForward 6.0 tools and may be installed during the FastForward 6.0 installation. The Pervasive drivers may also be installed by running SETUP.EXE from the ODBC directory within the FastForward 6.0 tools download. SQL Server ODBC drivers are installed with SQL Server 7 Client Utilities.

Prior to running *Integrity Wizard*, Data Source Names (DSN) need to be established for both the source system and application databases through ODBC Data Source Administrator. This section provides step-by-step instructions for creating the DSNs.

Creating Data Source Names — Pervasive ODBC-32 Drivers

The following steps should be followed to create the DSNs for Solomon IV 2.06 source databases:

1. Open *ODBC Data Source Administrator* through Windows *Control Panel*. Select the **User DSN** or **System DSN** tab, then click **Add**.



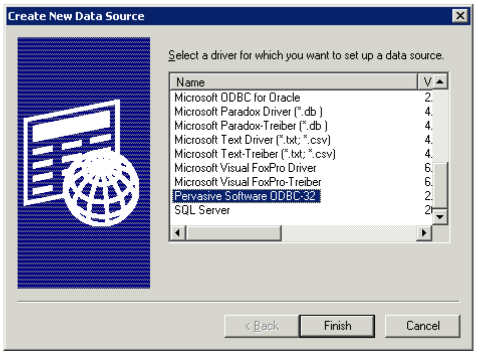
System Data Sources:

Name	Driver
CRSS	SQL Server
ECDCMusic	Microsoft Access Driver (*.mdb)
LocalServer	SQL Server
Xtreme Sample Database	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Figure 6: ODBC Data Source Administrator

2. Select the Pervasive Software ODBC-32 driver, then click **Finish**.



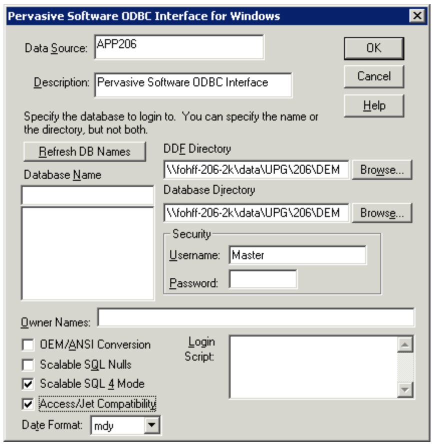
Create New Data Source

Select a driver for which you want to set up a data source.

Name	V
Microsoft ODBC for Oracle	2.
Microsoft Paradox Driver (*.db)	4.
Microsoft Paradox-Treiber (*.db)	4.
Microsoft Text Driver (*.txt; *.csv)	4.
Microsoft Text-Treiber (*.txt; *.csv)	4.
Microsoft Visual FoxPro Driver	6.
Microsoft Visual FoxPro-Treiber	6.
Pervasive Software ODBC-32	2.
SQL Server	2

Figure 7: Create New Data Source

3. Complete the *Pervasive Software ODBC Interface for Windows* screen as described below.



Pervasive Software ODBC Interface for Windows

Data Source: APP206

Description: Pervasive Software ODBC Interface

Specify the database to login to. You can specify the name or the directory, but not both.

Database Name

DDF Directory: \\fohff-206-2k\data\UPG\206\DEM

Database Directory: \\fohff-206-2k\data\UPG\206\DEM

Security

Username: Master

Password:

Owner Names:

☐ OEM/ANSI Conversion

☐ Scalable SQL Nulls

☑ Scalable SQL 4 Mode

☑ Access/Jet Compatibility

Date Format: mdy

Login Script:

Figure 8: Pervasive Software ODBC Interface for Windows

4. Create Data Source Names for both the application and the system source databases by entering a name for the source application database DSN in **Data Source**. Naming conventions for DSNs and their corresponding descriptions should be reflective of the source application and system databases for reference purposes.
5. Define the paths to the **DDF Directory** and the **Database Directory** by clicking **Browse** at the DDF Directory line Browse to the folder where that database's DDFs are located, highlight any of the DDF files and click **Open**. This should populate both the DDF Directory and the Database Directory connection lines. Connections can also be created through Network Neighborhood and pasted into these lines.

   Important: Do not used mapped network drives to connect to the DDF or Database Directories - full UNC pathing must be used. Also do not use the **Database Name** area.

6. Select **Scalable SQL 4 Mode** if a network connection is being utilized for ODBC. If Pervasive.SQL 7 Workstation Edition is being used for the migration on a standalone workstation **Scalable SQL 4 Mode** should not be selected. The functionality of integrity checks in *Integrity Wizard* is impacted by the selection.
7. Select **Access/Jet Compatibility**.
8. Click **OK** to finish.
9. Repeat the process to create the system database DSN.

Pervasive.SQL 7 Workstation Edition

Pervasive.SQL 7 Workstation Overview

The FastForward 6.0 installation wizard includes the option to install Pervasive.SQL 7 Workstation Edition, providing the ability to migrate Solomon IV 2.06 databases on a standalone workstation. Pervasive.SQL 7 Workstation Edition may be installed during the standard installation or at a later time by running SETUP.EXE from the PVSW directory within the FastForward 6.0 tools download. Installing Pervasive.SQL 7 Workstation Edition also automatically installs the Pervasive 2.54 ODBC drivers.

The FastForward 6.0 tools operate the same as if running the migration on the server. However, there are differences with respect to source database preparation, ODBC setup, and database repair. The following sections explain the modifications that need to be made to perform the migration using Pervasive.SQL 7 Workstation Edition.

Important: There are compatibility issues with Solomon IV 2.06 and Pervasive.SQL 7 Workstation Edition on the same workstation due to DLL conflicts. In order to utilize the FastForward 6.0 tools with Pervasive.SQL 7 Workstation Edition, install the tools on a workstation **not used to run** Solomon IV 2.06. Copy the source database files from their existing location to a directory on the workstation where FastForward 6.0 and Pervasive.SQL 7 are installed.

Since Solomon IV 2.06 cannot co-exist with Pervasive.SQL 7, some complexity is introduced with respect to resolving database conditions identified by *Integrity Wizard*. See “Resolving Database Error Conditions” on page 35.

Pervasive.SQL Workstation ODBC Setup

When creating the DSN through *ODBC Data Source Administrator* for a Solomon IV 2.06 database, **do not** select the option **Scalable SQL 4 Mode** in the *Pervasive Software ODBC Interface for Windows*. When checked, this option restricts the application from login using Scalable SQL 3.01 features which are needed to assure compatibility with the existing Solomon IV 2.06 Integrity Checks in Workstation Edition.

Workstation Edition Configuration

When the database is being migrated on a standalone workstation, Pervasive.SQL 7 Workstation Edition should be used. This scenario is appropriate when the source database is a Solomon IV 2.06 Single User edition or when NT servers are not available to perform the migration. Verify that the workstation running the migration has adequate disk space, approximately four times the size of the source database. When using this configuration, some of the information typically displayed on the *Integrity Wizard Analysis Report* will not be available since there is no path to the source Solomon IV application. Information obtained from the *Integrity Wizard* analysis will continue to display on the report.

Software Requirements

The following should be installed on the workstation:

- FastForward version 6.0
- Pervasive.SQL 7 Workstation Edition
- Pervasive.SQL ODBC (installs with Pervasive Workstation)
- Solomon IV version 6.0
- SQL Server 2000

Pervasive.SQL 7 — Setup Utility Modifications

Once Pervasive.SQL 7 is installed, modify the settings in the Pervasive.SQL 7 *Setup Utility* as described below. These settings modifications will not only increase performance, but are also needed to successfully complete execution of the integrity checks in *Integrity Wizard*.

File Settings

- Open Files: 2000
- Handles: 2000

Memory Resources

- Cache Allocations: 10000
- Communications Buffer Size: 40
- Extended Operation Buffer Size: 10000

System Resources/Directories

- I/O Threads: 20
- Worker Threads: 10

Converting Scalable Views from 3.01 to 4.0

Pervasive modified the internal format of the View.DDF with the release of Pervasive.SQL 7. Solomon IV 2.06 databases contain two views, Ratio Analysis and Balance Sheet, which are no longer being used. These views were originally used by early releases of Solomon IV in conjunction with the financial report writing module, Financial Report Writer. When FRx was introduced, these two views were not removed to allow clients using Financial Report Writer continued functionality. Pervasive.SQL 7 includes a utility VCONV32.EXE to update these views. The utility rebuilds the VIEW.DDF by using the foundation information in the original VIEW.DDF file and also updates the pointers in the other DDF files.

Note: The VIEW.DDF file may not exist in some databases. If the database never required Balance Sheet or Ratio Analysis and no other custom views were created, there will not be a VIEW.DDF file in the database directory. Therefore, you do not need to run the VCONV32 migration utility (please skip the step "Converting Scalable Views from 3.01 to 4.0").

> **These views must be migrated from the 3.01 to the 4.0 format prior to running the migration utilities. Any custom views must also be migrated. The process is as follows:**

1. Open the subdirectory where the source application database is installed.
2. Create a directory named SAVEVIEW in the SOL4/DB subdirectory.

3. Copy all DDFs to this new subdirectory. This will provide a backup of the original DDFs in the event that the process does not complete successfully. VCONV32.EXE does not modify the files in the SAVEVIEW subdirectory, but will reference them as a source for rebuilding VIEW.DDF and for updating the other DDF files.
4. Delete the VIEW.DDF file from the subdirectory where the application database is installed. VIEW.DDF will be rebuilt by VCONV32.EXE.
5. Convert the views using VCONV32.EXE that is distributed with Pervasive.SQL 7 and is installed in the PVSW\BIN subdirectory. Click **START**, then use the **RUN** command to convert the views using the following commands:

   - VCONV32 –dd:\soldb\app – fd:\soldb\app – vd:\soldb\app\saveview BalanceSheet
   - VCONV32 –dd:\soldb\app – fd:\soldb\app – vd:\soldb\app\saveview Ratio Analysis

   Note: The "–dd", "–fd", and "–v" command switches pass the correct location of the DDF and DAT files to the utility. The above commands assume that the database files (DAT and DDF) are stored in D:\SOLDB\APP. Modify as necessary for the current configuration.

Checklist for using Pervasive.SQL 7 Workstation Edition

X		Pervasive.SQL 7 Workstation Edition Checklist
	1.	Determine the size of the source database files and the amount of available disk space available on the workstation being used for the migration. The amount of available disk space should be approximately four times the size of the source database.
	2.	Verify that Solomon IV 2.06 is not installed on the workstation, then install Pervasive.SQL 7 Workstation Edition engine during the FastForward 6.0 installation or run SETUP.EXE from the PVSW directory within the FastForward 6.0 tools download.
	3.	Modify Pervasive.SQL 7 Workstation Edition settings using the Pervasive Setup Utility.
	4.	Verify Solomon IV version 6.0 and SQL Server 2000 are also installed on the workstation.
	5.	Copy the Solomon IV 2.06 application and system databases to a designated subdirectory on the workstation being used for the migration.
	6.	Create the DSNs through Pervasive ODBC Administrator. Do not check **Scalable SQL 4 Mode**.
	7.	Update existing views in Solomon IV 2.06 databases: ~~a.~~Open the subdirectory where the source application database is installed. ~~b.~~Create a directory named SAVEVIEW. ~~c.~~Copy all DDFs to this new subdirectory.

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

X		**Pervasive.SQL 7 Workstation Edition Checklist**
		~~d.~~Delete the VIEW.DDF file from the subdirectory where the application database is installed. ~~e.~~Convert the views using VCONV32.EXE that is distributed with Pervasive.SQL.
	8.	Use FastForward *Integrity Wizard* and FastForward *Transformation Wizard* to migrate the database to Solomon IV version 6.0.

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Resolving Database Error Conditions

To resolve some database error conditions, it may be necessary to make modifications to the source database from the Solomon IV 2.06 application. For example, some resolutions require that a screen in Solomon IV 2.06 be accessed in Initialize Mode to make the necessary modification. Since Pervasive.SQL 7 Workstation Edition and Solomon IV 2.06 cannot exist on the same workstation, a separate workstation running Solomon IV 2.06 is needed in order to correct database conditions that require login to Solomon IV 2.06.

When copying the database to a different workstation, only copy the *.DAT files. The DDF files are modified in order to run FastForward 6.0 with Pervasive.SQL 7 Workstation Edition, and therefore errors are received during attempts to login to Solomon IV 2.06 if the DDF files are copied with the database. Once the corrections have been made, the *.DAT files should be copied back to the workstation running FastForward.

If the modifications can be made using SQL Scope they can be made directly to the copy of the database being analyzed. However, you will need to utilize the version of SQL Scope distributed with Pervasive.SQL 7 Workstation Edition. When logging into the database, use the Database Path instead of Database Names to connect to the source database if named databases were not set up under the Pervasive Setup Utility, Maintain Named Databases.

Migrating to Select Edition

Select Edition Migration Overview

The primary issue with migrations to Solomon IV Select Edition is the difference between the Flexkey Definitions in Select and Premier. The table structure is identical for both editions; however, the Select Edition has restrictions in the number of characters allowed for subaccount and task ID, as well as the number of Flexkey segments allowed for all Flexkey fields.

The FastForward 6.0 tools will facilitate migration to Select Edition in the following two scenarios:

1. In the first scenario, the source database Flexkey definitions naturally conform to the Select Edition Flexkey maximum segment and maximum character requirements. The migration can proceed without post-migration modifications.

   Following are the Select Edition Flexkey requirements:

Flexkey	Maximum Characters	Maximum Segments
Subaccount	6	3
Customer ID	15	1
Vendor ID	15	1
Inventory ID	30	1
Employee ID	10	1
Task ID	24	1

2. In the second scenario, the number of characters in all Flexkeys conform to the requirements for Select, but the number of segments in one or more of the source Flexkeys exceeds the number allowed in Select. In this scenario, a post-migration utility is provided to modify the number of segments to conform to Select criteria. More detail on the post-migration process is provided under the section, “Flexkey Field Segments Exceed Select Edition Requirements” on page 39.

Important: FastForward 6.0 will not support migrations to Select Edition where the number of characters in the Flexkey exceeds the maximum characters allowed for Select Edition.

Integrity Wizard Analysis Report

The *Integrity Wizard Analysis Report* includes a section labeled “Select/Premier Edition Qualifiers.” The report analyzes the Select Edition criteria and indicates whether the Flexkey definitions in the source database meet the criteria. If any of the criteria return a result of Fail, the database cannot be migrated to Select. If a Warning is returned, the database can be migrated, however, the post-migration script must be run on the migrated database to modify the Flexkey table. These modifications are described in the following section.

Flexkey Field Segments Exceed Select Edition Requirements

If the source database meets the maximum character requirements, but the number of segments exceeds the number allowed for one or more Flexkeys, the database can still be migrated.

In this situation, the database should be migrated using standard procedures. After the database has been successfully migrated, one of the scripts provided in the Utilities subdirectory within the FastForward 6.0 tools download should be run against the migrated database.

These scripts modify the contents of the Flexkey table in the migrated database. These scripts force all Flexkey fields to one segment of appropriate length. After the script is executed, it will be necessary to log in to Solomon IV version 6.0 and make final modifications in *Flexkey Definition* (21.320.00) and potentially in *Flexkey Table Maintenance* (21.330.00).

For example, if three segments are desired for the Subaccount Flexkey, the segments will need to be defined in *Flexkey Definition* (21.320.00), since the script moves all characters to one segment. In addition, *Flexkey Table Maintenance* (21.330.00) would need to be modified to coincide with the new segment definitions.

Following are descriptions for each of the scripts located in the Utilities directory within the FastForward 6.0 tools download:

- The SELECT EDITION ALL.SQL script updates the FLEXDEF table, by setting the Subaccount Flexkey (Fieldclass 001) to a single, 6 character field. All other Flexkey fields are forced to 1 segment of appropriate length.
- The SELECT EDITION SUBACCOUNT.SQL script only updates the Subaccount Flexkey as described above. This script should be used in situations where all Flexkeys, other than Subaccount, meet the Select Edition criteria.
- The SELECT EDITION OTHER.SQL script updates all Flexkeys, except Subaccount. This script should be used in situations where any Flexkey, other than Subaccount, does not meet the segment criteria for Select.

  Note: These scripts will only need to be executed if migrating to Select Edition.

Since all the scripts set the specified Flexkeys to one segment and remove segment descriptions and edit masks, it will be necessary to login to Solomon IV version 6.0 and make appropriate modifications to segment definitions in *Flexkey Definition* (21.320.00). In addition, modifications may need to be made in *Flexkey Table Maintenance* (21.330.00).

The following example describes a typical scenario.

Select Edition Migration Example

Licensed modules: General Ledger, Accounts Payable, and Accounts Receivable:

Before SELECT EDITION.SQL			**After** SELECT EDITION.SQL		
Subaccount Flexkey: 4 Segments			**Subaccount Flexkey:** 1 Segment		
Seg #	**# of Char**	**Desc**	**Seg #**	**# of Char**	**Desc**
1	2	Region	1	6	
2	1	Division			
3	1	Unit			
4	2	Dept			
Vendor Flexkey: 2 Segments			**Vendor Flexkey:** 1 Segment		
1	2	Region	1	15	
2	13	Name			
Customer Flexkey: 1 Segments			**Customer Flexkey:** 1 Segment		

1	10	Name	1	10	

Using Integrity Wizard

Integrity Wizard Overview

FastForward *Integrity Wizard* analyzes the existing Solomon database and prepares it for upgrade. It uses an industry standard wizard interface to examine the source database and report on results. The primary purpose for *Integrity Wizard* is to prevent irregular database conditions that exist in the Solomon IV 2.06 source database from passing through to the migrated Solomon IV version 6.0 database. The database analysis is accomplished through an extensive series of queries that identify conditions in the database that will either prevent the migration from executing successfully or may have a negative impact in the migrated database.

In order to complete a successful upgrade, a preliminary (or “test”) migration should be scheduled well in advance of the targeted final (or “live”) migration date. Running a preliminary test migration provides several benefits:

- Ability to estimate the time needed to run the actual migration
- Assessment of the integrity of the data within the database
- Ability to research and estimate the time needed to repair error conditions
- Analysis of database customizations and custom reports that will potentially need to be recreated to complete the migration

Microsoft Business Solutions recommends that the preliminary migration be performed offsite with a copy of the client's database. A current copy of the database will then be needed to run the final migration if the live migration is not being performed at the client site. Data from the preliminary migration will provide a good estimate of the time needed to complete the live migration, therefore client expectations can be set accordingly.

Version 6.0 Defaults

Integrity Wizard includes screens to define default values, by module, for data that is not maintained in the Solomon IV 2.06 source databases or data that needs to be re-stated for migration purposes. For example, earlier Solomon IV products did not include a Pre-Payment Account and Subaccount in Accounts Receivable. *Integrity Wizard* provides an area to define these values, as well as defaults for other modules available for upgrade. The default values entered are written to a specific file, SOLUPTAB.CNV that is utilized by *Transformation Wizard* during the migration process to set the specified values in the Solomon IV version 6.0 database. See "Default Value Definitions Checklist" on page 57 for a listing of the default values that will need to be defined. The checklist is a tool that should be incorporated into the migration planning process to assist in collecting essential information prior to the upgrade. Descriptions of the default fields contained in *Integrity Wizard* are included in "Integrity Wizard Defaults and Module Specifications" on page 94.

Database Analysis

A primary objective of *Integrity Wizard* is to scrutinize the current database to identify conditions that will negatively impact migration of the data to Solomon IV v6.0. This is accomplished by executing a suite of specially designed SQL (Structured Query Language) scripts that check the database for integrity issues such as an out-of-balance trial balance, batches which are unreleased or unbalanced or unposted, orphan transaction records, fields with invalid null values and invalid date values. Each integrity check will return a status of No Errors, Warning, or Fatal Error. Conditions flagged with a status of Warning indicate situations that warrant review and possible refinement, but will not prevent the migration from proceeding. Fatal Errors however, must be reviewed and corrected before a final live migration can be completed. *Transformation Wizard* will allow what is referred to as a “Preliminary Migration” to be run against the database while Fatal Errors remain in the source database. See “Preliminary ” on page 230 for more information on running test migrations.

Although the Integrity Wizard’s check suite is comprehensive, and captures the vast majority of irregular database conditions, there may be occasional circumstances where a database irregularity is not identified until the actual migration process is initiated through *Transformation Wizard*. In these cases it will be necessary to review the error logs created by *Transformation Wizard* to resolve the database irregularity.

Database Repair

Integrity Wizard has the capability to systematically repair a number of error conditions. However, many conditions require manual review and repair. In these cases, *Integrity Wizard* has the power to export reference information to Microsoft Excel for each integrity check that returns an error. Used by a trained database professional, these worksheets provide the information necessary to investigate and repair database issues. See “Integrity Check Information” on page 131 for a listing of integrity checks by module, including explanations of the integrity checks and how to resolve error conditions.

Integrity Wizard Analysis Report

The *Integrity Wizard Analysis Report* is available at the completion of the *Integrity Wizard* analysis. The report includes information derived from the *Integrity Wizard* analysis as well as information read from the source Solomon IV application subdirectory.

The *Integrity Wizard Analysis Report* is intended to assist in the overall assessment of the work effort needed to complete the upgrade. For example, the report includes a section on customizations, which includes information on customizations for screens, reports, custom tables, etc. This type of information facilitates a comprehensive assessment of the tasks needed to complete the upgrade. See “Integrity Wizard Analysis Report” on page 128 for more information on using this component of *Integrity Wizard*.

Source Database Preparation Checklist

This checklist is intended to provide a guideline for preparing the source database for migration. Some of the Action Item categories, such as Batch Status, apply globally to the source database. However, specific preparations are needed for certain modules and are so noted. If a specific module is not listed, only the global preparation criteria apply.

Client database:			**Solomon Version:**
Completed by:			**Date:**
X		**Action Item**	**Comments/Results**
	1.	Verify version of source database (Should be Solomon IV 2.06).	
	2.	~~□~~If Project Series is being migrated verify that version B.02 SP1 is installed for Solomon IV 2.06.	
	3.	Create a restorable backup of the source database.	
	4.	Create ODBC connections to Source Database through ODBC Data Source Administrator.	
	5.	Purge historical data if necessary.	

Formatted: Bullets and Numbering

Client database:			**Solomon Version:**
Completed by:			**Date:**
X		**Action Item**	**Comments/Results**
	6.	**Batch Statuses**	
		Release all batches through the current period (the only exception is Accounts Payable batches created from Purchasing Receipts).	
		Post all batches through the current period.	
		Release all batches entered to future periods.	
		Repair any Suspended or Partially Released batches.	
	7.	**Project Series** — Release, post, or delete all open documents or batches in the following modules:	
		• Travel & Expense	
		• Time and Expense for Projects	
		• Project Charge Entry	
		• Budget	

Client database:			**Solomon Version:**
Completed by:			**Date:**
X		**Action Item**	**Comments/Results**
	8.	**Project Series Flexible Billings —** Approve, print and post, or delete all invoice drafts.	
	9.	**Project Series —** The period must be closed. Any warnings that appear should be resolved before proceeding. No processing should take place (even in future period) after the period has been closed.	
	10.	**Project Series —** The period must be the same as the current period in General Ledger.	
	11.	**Inventory**	
		Verify physical inventory valuation is accurate and make any necessary adjustments.	

Client database:			**Solomon Version:**
Completed by:			**Date:**
X		**Action Item**	**Comments/Results**
		Inventory — Lot/Serial Numbers Verify validity of items with lot/serial numbers. *Transformation Wizard* will only migrate lot/serial detail on items that are indicated as Lot or Serial Tracked in *Inventory Items* (10.250.00) and have the Serial Number Assignment set to When Received.	
	12.	**Order Processing**	
		Verify all open orders have one of the following statuses: S — Sales Order I — Invoice O — Open Order Q — Quote	
		No orders have a status of Credit Hold (R) or Admin Hold (H).	

Client database:			**Solomon Version:**
Completed by:			**Date:**
X		**Action Item**	**Comments/Results**
		All cash sales orders (CA) must be closed since open CA orders will not be migrated.	
		All credit memos (CM) must be closed since open credit memos will not be migrated.	
	13.	**Service Series**	
		All sales order transactions created from Service Call must be closed since they will not be migrated. **Warning**: Failure to follow this step could result in performance and integrity issues within your database.	

Client database:			**Solomon Version:**
Completed by:			**Date:**
X		**Action Item**	**Comments/Results**
		All service calls with labor, material, or other detail lines should be processed through to Order Processing, Accounts Receivable, and Inventory. This is not mandatory but is recommended to mitigate risk.	
		All service order transactions that came from Service Contract must be closed since open service orders from Service Contract will not be migrated.	

Source Database Reports Checklist

Print the following reports for the modules being migrated from the source database and save for post-migration validation with the migrated database.

Client Database:		**Solomon Version:**
Completed by:		**Date:**
Module	**X**	**Report**
General Ledger		*Trial Balance* (01.610.00) report for the current period
		Trial Balance (01.610.00) report for the prior fiscal year end
Accounts Payable		*Aged AP* (03.680.00) report — summary format
		Vendor Trial Balance (03.650.00) — suggest printing to file due to length of report
Accounts Receivable		*Customer Trial Balance* (08.620.00) — suggest printing to file due to length of report
		Aged AR (08.610.00) report — summary format
Project Controller		*Project Task Summary* (PA.06.600.00) for current period (if flexible columns for the report have been defined)

Client Database:		**Solomon Version:**
Completed by:		**Date:**
Module	**X**	**Report**
		Project Transactions (PA.06.700.00) for the last period and the current period; advisable to print to file or select by transaction date range for both reports to minimize the number of pages printed
Payroll		*Labor Distribution by Charged or Home Subaccount* (TM.030.00) in Time and Expense for Projects for the current period; advisable to print to file or select by a few employees to minimize the number of pages printed
Inventory		*Earnings & Deductions* (02.670.00)
		Inventory Valuation (10.620.00) report
		Inventory Trial Balance (10.630.00) report — current period only
Purchasing		*Purchase Order Register* (04.620.00)
		Receipts Register (04.650.00)
Bill of Material		*Bill of Material Lists* (11.600.00) — detail format — suggest printing to file due to length of report
		Production Analysis (11.660.00) — detail format — suggest printing to file due to length of report

Default Value Definitions Checklist

Integrity Wizard requires various default values for each module being migrated. This checklist includes the default fields required by each module. See “Integrity Wizard Defaults and Module Specifications” on page 94 for descriptions of each default item.

Client Database:			**Solomon Version:**
Completed by:			**Date:**
Module	**X**	**Default Item**	**Default Value**
Dates		Fiscal Year ends in its calendar year	Yes ____ No ____
		Earliest Valid Year	
		Latest Valid Year	
		Pause *Integrity Wizard* after date check	Yes ____ No ____
General Ledger		Company ID	
		Actual Ledger ID	
		Budget Ledger ID	
		Memo1 Ledger ID	

Client Database:			**Solomon Version:**
Completed by:			**Date:**
Module	**X**	**Default Item**	**Default Value**
		Memo2 Ledger ID	
Accounts Payable		Default Vendor Purchase Type (Select One)	GN – Goods Non-Inventory DL – Description Line GI – Goods Inventory SE – Services for Expenses SP – Services for Project
Accounts Receivable		Pre-Payment Acct/Sub	
		Credit Memo Terms	
		Default Terms	
Inventory Lot/Serial		Fixed Type (Select One)	E – Enterable C – Constant
		Issue Method (Select One)	S – Sequential E – Expiration F – FIFO L – LIFO
		Shelf Life (If Issue Method = E)	_____ Days
		Fixed Length of LSN Prefix	

Client Database:			**Solomon Version:**
Completed by:			**Date:**
Module	**X**	**Default Item**	**Default Value**
		Value of Prefix (If Used)	
		Length of Lot Number	
		Number Value	
		Default Tax Category	
Inventory Acct/Sub		AP Clearing Account	Account __________________ Subaccount ________________
		AR Clearing Account	Account __________________ Subaccount ________________
		IN Transit Account	Account __________________ Subaccount ________________
		STD Cost Reval Account	Account __________________ Subaccount ________________

Client Database:			**Solomon Version:**
Completed by:			**Date:**
Module	**X**	**Default Item**	**Default Value**
		Material Ovrhd Offset	Account ___________________ Subaccount _________________
		COGS Account	Account ___________________ Subaccount _________________
		Inventory Account	Account ___________________ Subaccount _________________
		Sales Account	Account ___________________ Subaccount _________________
		Purchase Price Variance Account	Account ___________________ Subaccount _________________
		STD Cost Variance Account	Account ___________________ Subaccount _________________
Inventory Item		Material Type	

Client Database:			**Solomon Version:**
Completed by:			**Date:**
Module	**X**	**Default Item**	**Default Value**
		Transaction Status Code (Select One)	AC – Active NP – No Purchase NU – No Usage IN – Inactive OH – On Hold
			On Hold
			No Usage
			Inactive
			No Purchase
		Inventory Item Unit	
		Material Overhead Calculation (Select One)	R – When Received U – When Used N – None
			When Used
		Dec Place Precision — Cost (select a value between 0 and 6)	Select a Value between 0 and 6
Purchasing		Freight Account/Subaccount	Account__________________ Subaccount_______________

Client Database:			**Solomon Version:**
Completed by:			**Date:**
Module	**X**	**Default Item**	**Default Value**
		Vouchering Stage (Select One)	All Fully Vouchered All Completely Unvouchered Date Based: ___/___/___ AP Doc Based
			All Completely Unvouchered
			Date Based
			AP Doc Based
Order Management Descr		Order Management Edition (Select One)	Standard ___ OM Plus ____
		Default Ship Via ID	
		Default Ship Via Description	
		Default Description Inventory ID	
		Default Description Inventory ID: Service Orders	

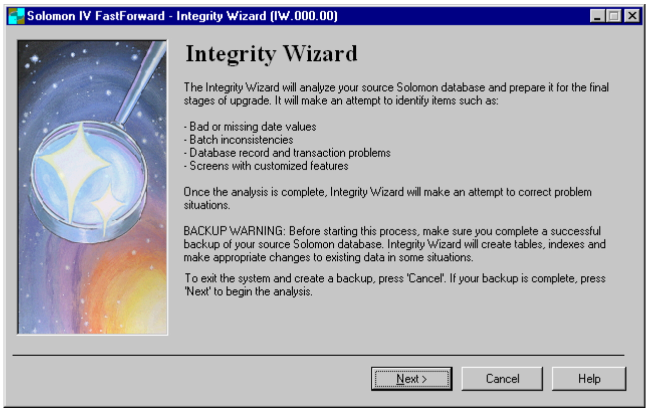
Client Database:			**Solomon Version:**
Completed by:			**Date:**
Module	**X**	**Default Item**	**Default Value**
		Default Salesperson Method	Customer Ship-to Address User Both User and Customer Ship-to Address No Default
			User
			Both User and Customer Ship-to Address
			No default
Order Management Acct/Sub		Inventory Scrap Account	Account ________________ Subaccount ______________
		Error Account	Account ________________ Subaccount ______________

FastForward Integrity Wizard Step-by-Step

After preparing the source database and reviewing the checklists to ensure all necessary actions have been taken, the next step is to run *Integrity Wizard*. The following instructions explain how to use *Integrity Wizard*.

> **Launch FastForward *Integrity Wizard* from the Solomon IV program group:**

1. *Integrity Wizard* begins with a brief introduction and a Backup Warning. A restorable backup of the source database should be created before proceeding.



Solomon IV FastForward - Integrity Wizard (IW.000.00)

Integrity Wizard

The Integrity Wizard will analyze your source Solomon database and prepare it for the final stages of upgrade. It will make an attempt to identify items such as:

- Bad or missing date values
- Batch inconsistencies
- Database record and transaction problems
- Screens with customized features

Once the analysis is complete, Integrity Wizard will make an attempt to correct problem situations.

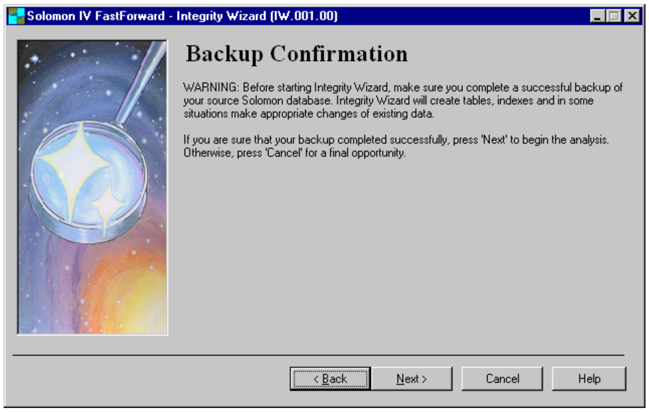
BACKUP WARNING: Before starting this process, make sure you complete a successful backup of your source Solomon database. Integrity Wizard will create tables, indexes and make appropriate changes to existing data in some situations.

To exit the system and create a backup, press 'Cancel'. If your backup is complete, press 'Next' to begin the analysis.

Figure 9: Integrity Wizard (IW.000.00)

- Click **Next** if a restorable backup of the source database has been created.
- Click **Cancel** to exit *Integrity Wizard* and create a restorable backup of the source database before proceeding.

2. *Backup Confirmation* appears next and reiterates the necessity of creating a restorable backup. *Integrity Wizard* creates new tables and indexes, and in cases of irregular records, may make changes to existing data. Therefore, it is critical that a restorable backup of the source database is available.



Backup Confirmation

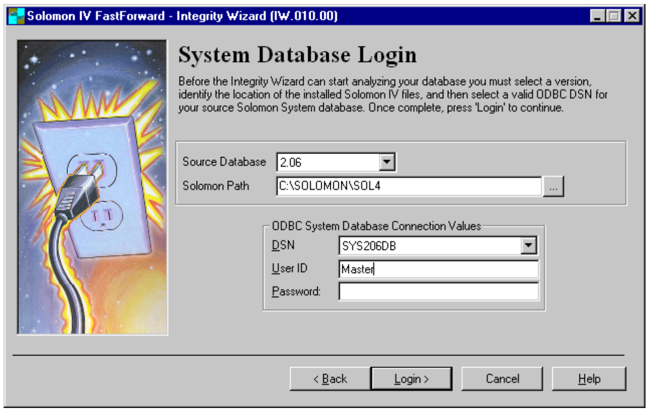
WARNING: Before starting Integrity Wizard, make sure you complete a successful backup of your source Solomon database. Integrity Wizard will create tables, indexes and in some situations make appropriate changes of existing data.

If you are sure that your backup completed successfully, press 'Next' to begin the analysis. Otherwise, press 'Cancel' for a final opportunity.

Figure 10: Integrity Wizard (IW.001.00) — Backup Confirmation

- Click **Next** to proceed with *Integrity Wizard*.
- Click **Cancel** to exit *Integrity Wizard* in order to create a restorable backup.

3. Select the source database version and enter the path to the Solomon IV source application. Enter the login information for the source system database.



Solomon IV FastForward - Integrity Wizard (IW.010.00)

System Database Login

Before the Integrity Wizard can start analyzing your database you must select a version, identify the location of the installed Solomon IV files, and then select a valid ODBC DSN for your source Solomon System database. Once complete, press 'Login' to continue.

Source Database: 2.06

Solomon Path: C:\SOLOMON\SOL4

ODBC System Database Connection Values

DSN: SYS206DB

User ID: Master

Password:

Figure 11: Integrity Wizard (IW.010.00) — System Database Login

4. Select the version of the source database from the **Source Database** drop-down list. If the migration is for a source Solomon IV 2.06 database using a standalone workstation and Pervasive.SQL 7 Workstation Edition, select the option, 2.06 Workstation.

5. Enter the path of the source Solomon IV installation in **Solomon Path**. This is the path where your Solomon IV version 2.06 Parent is located. *Integrity Wizard* uses this path to locate information for the *Integrity Wizard Analysis Report*. It is not necessary to provide a correct path, however, if you choose not to provide a correct path, an error will occur later in *Integrity Wizard*. Also, the Customized Executables, Customized Reports, SOLOMON.INI, and Menu Customizations sections will be blank.

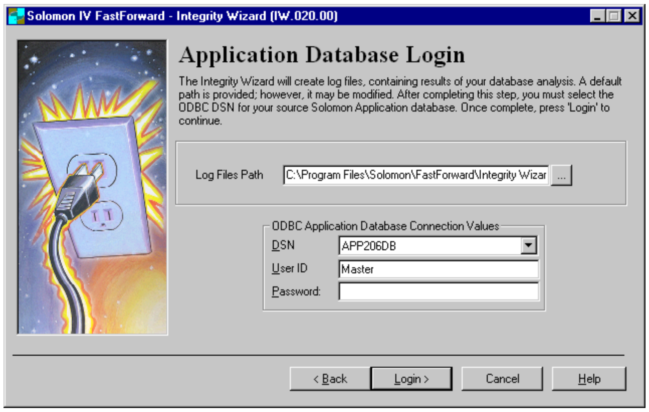
   Note: If you are running Pervasive.SQL 7 Workstation Edition on a standalone workstation, a valid path will not be available when migrating Solomon IV 2.06 databases, since Solomon IV 2.06 and Pervasive.SQL Workstation cannot co-exist on the same workstation.

6. Under ODBC System Database Connection Values, select the **DSN** that was created in ODBC Administrator for the source system database. Enter the appropriate **User ID**: Master for Solomon IV 2.06 databases. Enter a **Password** if security has been established on the source database.

7. Click **Login** to proceed.

   Note: Errors received after clicking **Login** are typically an indication that the ODBC connection is not configured correctly.

8. Enter the path for the *Integrity Wizard* log files and login information for the source application database.



Solomon IV FastForward - Integrity Wizard (IW.020.00)

Application Database Login

The Integrity Wizard will create log files, containing results of your database analysis. A default path is provided; however, it may be modified. After completing this step, you must select the ODBC DSN for your source Solomon Application database. Once complete, press 'Login' to continue.

Log Files Path: C:\Program Files\Solomon\FastForward\Integrity Wizar

ODBC Application Database Connection Values

DSN: APP206DB

User ID: Master

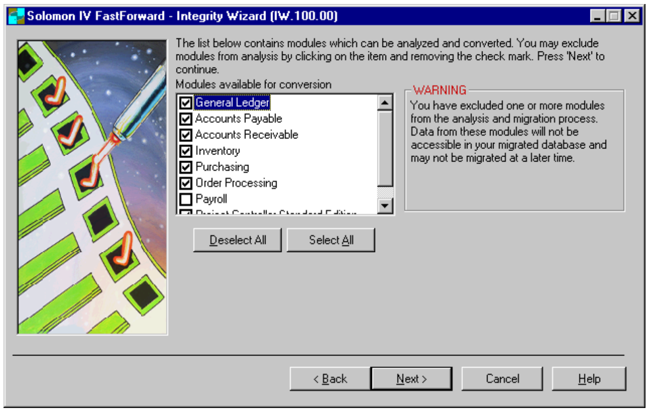
Password:

Figure 12: Integrity Wizard (IW.020.00) — Application Database Login

9. **Log Files Path** — Select the directory where the *Integrity Wizard* log files will be stored. When *Integrity Wizard* detects database error conditions, the result sets are stored in these Log Files.

10. Under ODBC Application Database Connection Values, select the **DSN** that was created in ODBC Administrator for the source application database. Enter the appropriate **User ID**: Master for Solomon IV 2.06 databases. Enter a **Password** if security has been established on the source database.
11. Click **Login** to proceed.

12.Select the modules to convert.



Solomon IV FastForward - Integrity Wizard (IW.100.00)

The list below contains modules which can be analyzed and converted. You may exclude modules from analysis by clicking on the item and removing the check mark. Press 'Next' to continue.

Modules available for conversion

- ☑ General Ledger
- ☑ Accounts Payable
- ☑ Accounts Receivable
- ☑ Inventory
- ☑ Purchasing
- ☑ Order Processing
- □ Payroll
- ☑ Project Controller Standard Edition

WARNING

You have excluded one or more modules from the analysis and migration process. Data from these modules will not be accessible in your migrated database and may not be migrated at a later time.

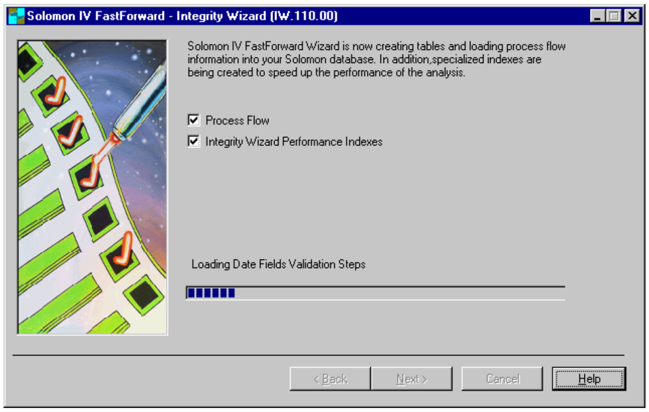
Figure 13: Integrity Wizard (IW.100.00) — Module selection

Important: *Modules available for migration* displays the modules that contain setup records in the source database and are available for migration. By default, all eligible modules are selected. Modules in this list may be excluded by clicking in the checkbox beside the module name. If a module is deselected, the Warning text displays. Modules that are deselected will not be analyzed by *Integrity Wizard* and will not be migrated by *Transformation Wizard*.

Important: Modules not available for migration only display when *Integrity Wizard* detects modules implemented in the source database but not supported for migration in the current release of FastForward. If this condition exists, the Warning displays, indicating that data from these modules will not be migrated and cannot be migrated at a later time.

After selecting the modules that are desired for the migration, click **Next** to proceed.

13. Next *Integrity Wizard* automatically creates specialized tables, process flow (integrity check scripts) and indexes in the source database.



Solomon IV FastForward Wizard is now creating tables and loading process flow information into your Solomon database. In addition,specialized indexes are being created to speed up the performance of the analysis.

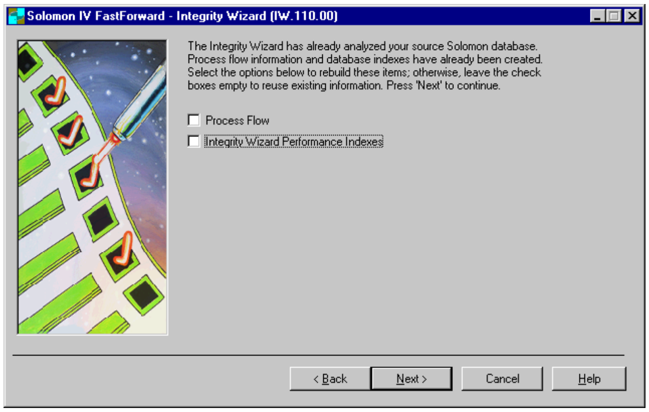
☑ Process Flow

☑ Integrity Wizard Performance Indexes

Loading Date Fields Validation Steps

Figure 14: Integrity Wizard (IW.110.00) — Creating Process Flow and Indexes

14. When running *Integrity Wizard* subsequent times against the same database, loading **Process Flow** and **Integrity Wizard Performance Indexes** will be optional.



The Integrity Wizard has already analyzed your source Solomon database. Process flow information and database indexes have already been created. Select the options below to rebuild these items; otherwise, leave the check boxes empty to reuse existing information. Press 'Next' to continue.

□ Process Flow

□ Integrity Wizard Performance Indexes

Figure 15: Integrity Wizard (IW.110.00) — Optional Process Flow and Indexes Load

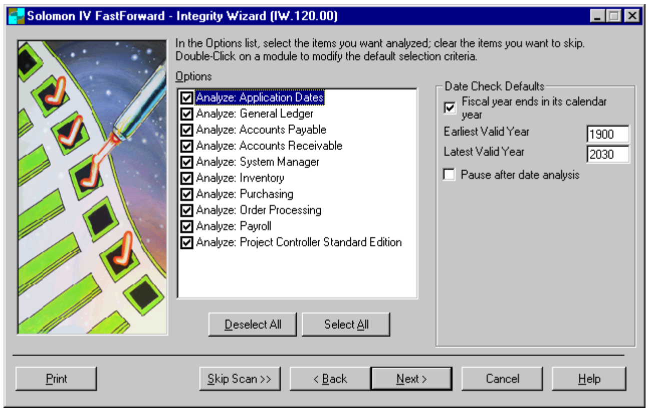
Reload Process Flow to:

Reset the **Status** values for the integrity checks to their original state of Pending and to select all integrity checks to process. After running an initial pass through *Integrity Wizard* each integrity check status reflects the results of the data analysis. In order to run a complete second pass on the database, select **Process Flow** to ensure that all checks are reselected for processing. This is important in situations where the database has been modified to correct error conditions after an initial pass through *Integrity Wizard*. Several of the integrity checks create temporary tables that contain summary data from multiple tables associated with a particular module. These temporary tables are then used by subsequent integrity checks for data analysis. If **Process Flow** is not reset, *Integrity Wizard* may return the same error since the summary tables were not rebuilt.

Reload Performance Indexes to:

Reload the indexes that are created by *Integrity Wizard* if indexes were dropped at completion of the initial pass through *Integrity Wizard*. *Integrity Wizard* includes an option to drop indexes after *Integrity Wizard* database analysis is completed. Dropping indexes is recommended in situations where the database will be used in a production environment.

15. Select database analysis criteria and enter Solomon IV v6.0 default values.



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

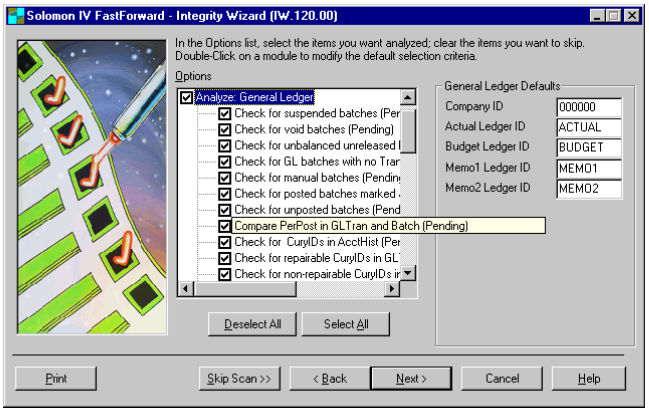
- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Date Check Defaults

- ☑ Fiscal year ends in its calendar year
- Earliest Valid Year: 1900
- Latest Valid Year: 2030
- ☐ Pause after date analysis

Figure 16: Integrity Wizard (IW.120.00) — Database analysis criteria and module defaults

- All modules selected for migration in the previous screen will be selected for analysis by default. Microsoft Business Solutions recommends leaving all options selected to ensure a complete review of the source database.
- Double-click on a module to display the specific checks within each module. (See Figure 25.) Specific checks within a module may be individually deselected. Integrity checks should only be deselected in situations where manual analysis is needed to verify corrections made to the database for a specific integrity check.
- Click **Print** to produce a list of all integrity checks and the corresponding status of each.



In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: General Ledger
  - ☑ Check for suspended batches (Per
  - ☑ Check for void batches (Pending)
  - ☑ Check for unbalanced unreleased I
  - ☑ Check for GL batches with no Trar
  - ☑ Check for manual batches (Pendin
  - ☑ Check for posted batches marked .
  - ☑ Check for unposted batches (Pend
  - ☑ Compare PerPost in GLTran and Batch (Pending)
  - ☑ Check for CuryIDs in AcctHist (Per
  - ☑ Check for repairable CuryIDs in GL
  - ☑ Check for non-repairable CuryIDs ir

General Ledger Defaults

Company ID	000000
Actual Ledger ID	ACTUAL
Budget Ledger ID	BUDGET
Memo1 Ledger ID	MEMO1
Memo2 Ledger ID	MEMO2

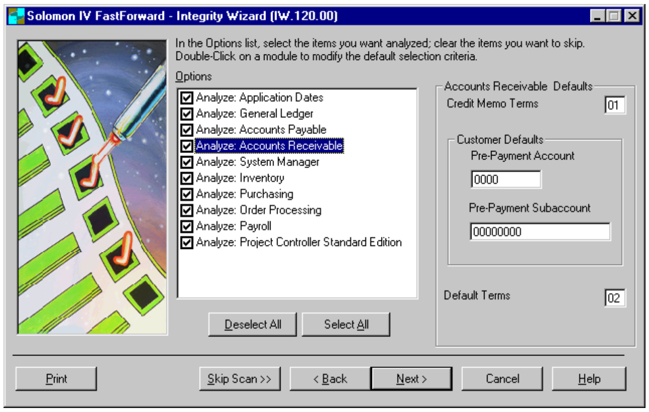
Figure 17: Integrity Wizard (IW.120.00) — Integrity check criteria

16. The status of each integrity check is displayed by highlighting the desired check. At the start of an initial scan, the default status value will be Pending for all checks. After the initial scan, the status changes according to the conditions that *Integrity Wizard* identifies.

    Status values:

    - Completed — *Integrity Wizard* completed with no errors detected.
    - Deselected — The check is excluded from analysis.
    - Fatal Error — *Integrity Wizard* identified conditions that must be resolved prior to migrating the database.
    - Pending — The check is selected for analysis.
    - Repaired — *Integrity Wizard* detected error conditions and the option to automatically repair the condition was selected and completed successfully during the initial scan or after reloading process flow.
    - Warning — *Integrity Wizard* was run and conditions were identified that warrant investigation and possible modification but will not prevent the database from migrating.

17. Set default values for dates and modules for the migrated database.



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip.
Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Accounts Receivable Defaults

Credit Memo Terms 01

Customer Defaults

Pre-Payment Account 0000

Pre-Payment Subaccount 00000000

Default Terms 02

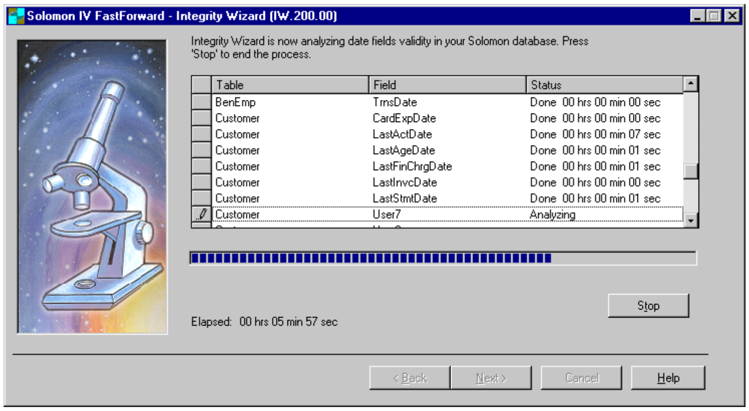
Figure 18: Integrity Wizard (IW.120.00) — Default values

Highlight Analyze: Application Dates or any of the modules to refresh the defaults frame and display the default options for the selection.

See “Integrity Wizard Defaults and Module Specifications” on page 94 for descriptions of the default options and recommendations for setting values. It is important that the default options are completed for all modules being migrated. The Default Values Information Checklist should be used as a tool to collect the default information prior to the migration process.

18. After defaults are defined, *Integrity Wizard* is ready to begin analyzing date fields to ensure the database contains valid date values.

    Although the date analysis is comprehensive, there are instances where a bad date is not detected by the SQL query because of invalid characters in the date field. This causes the potential of invalid dates not being identified. Invalid date values will cause an error during *Transformation Wizard*. In the event that *Transformation Wizard* identifies additional invalid dates, the error logs in *Transformation Wizard* will indicate the tables that need to be reviewed and modified in order for the migration to complete successfully.



Solomon IV FastForward - Integrity Wizard (IW.200.00)

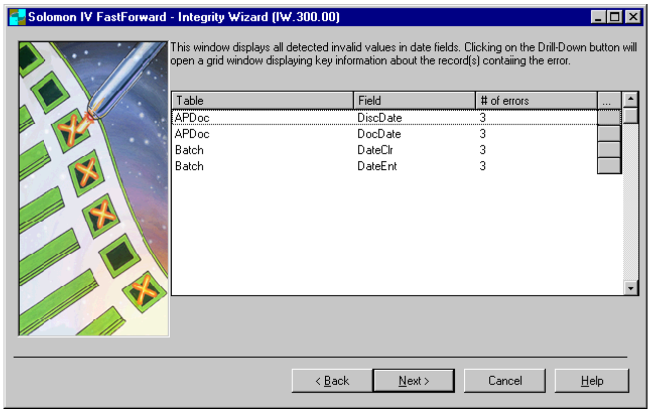
Integrity Wizard is now analyzing date fields validity in your Solomon database. Press 'Stop' to end the process.

Table	Field	Status
BenEmp	TrnsDate	Done 00 hrs 00 min 00 sec
Customer	CardExpDate	Done 00 hrs 00 min 00 sec
Customer	LastActDate	Done 00 hrs 00 min 07 sec
Customer	LastAgeDate	Done 00 hrs 00 min 01 sec
Customer	LastFinChrgDate	Done 00 hrs 00 min 01 sec
Customer	LastInvcDate	Done 00 hrs 00 min 00 sec
Customer	LastStmtDate	Done 00 hrs 00 min 01 sec
Customer	User7	Analyzing

Elapsed: 00 hrs 05 min 57 sec

Figure 19: Integrity Wizard (IW.200.00) — Date field analysis

19. After performing the Database Analysis, *Integrity Wizard* (IW.300.00) displays the results of the Date Field Analysis. Each line in the results grid represents a date field which was found to be problematic. The results grid displays the table name, field name, and the number of records which contained an invalid date.

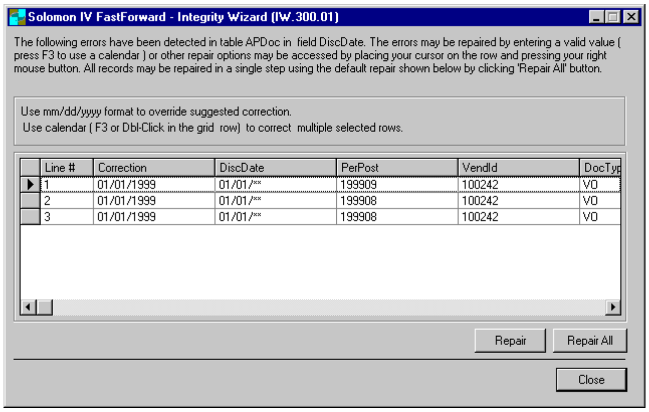


This window displays all detected invalid values in date fields. Clicking on the Drill-Down button will open a grid window displaying key information about the record(s) contaiing the error.

Table	Field	# of errors	...
APDoc	DiscDate	3	
APDoc	DocDate	3	
Batch	DateClr	3	
Batch	DateEnt	3	

Figure 20: Date field analysis results

20. Double-click on a line in the Date Field Analysis Results grid or click on the **Drill-Down** button in the right-most column of the grid to display the detail of the results for each particular date field. If valid corrections are suggested in the **Corrections** column of the result set, click on the **Repair All** button to change all of the date fields in the result set to the suggested dates. Corrections can be made one at a time using the **Repair** button to change just the current line. If no corrections are suggested, or changes to the suggested corrections need to be made, click on a field in the **Corrections** column and type in the correct date.



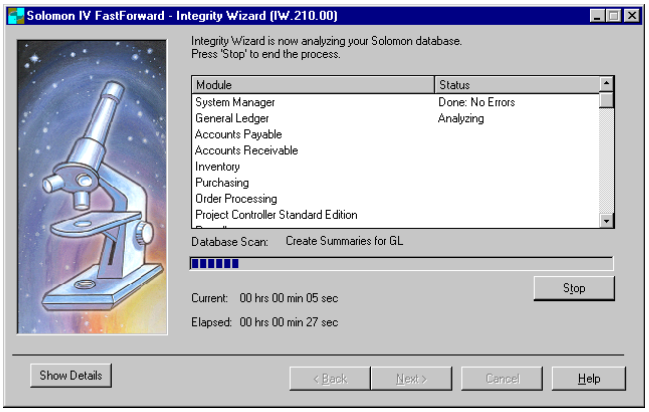
The following errors have been detected in table APDoc in field DiscDate. The errors may be repaired by entering a valid value (press F3 to use a calendar) or other repair options may be accessed by placing your cursor on the row and pressing your right mouse button. All records may be repaired in a single step using the default repair shown below by clicking 'Repair All' button.

Use mm/dd/yyyy format to override suggested correction.
Use calendar (F3 or Dbl-Click in the grid row) to correct multiple selected rows.

Line #	Correction	DiscDate	PerPost	VendId	DocTyp
1	01/01/1999	01/01/**	199909	100242	VO
2	01/01/1999	01/01/**	199908	100242	VO
3	01/01/1999	01/01/**	199908	100242	VO

Figure 21: Date field analysis result detail

21. *Integrity Wizard* automatically begins the remainder of the analysis when the date field analysis completes if the “Pause integrity check after date check” option was not selected in the Analyze: Application Dates default list. If the option was selected, click **Start** to begin analysis.



Solomon IV FastForward - Integrity Wizard (IW.210.00)

Integrity Wizard is now analyzing your Solomon database.
Press 'Stop' to end the process.

Module	Status
System Manager	Done: No Errors
General Ledger	Analyzing
Accounts Payable	
Accounts Receivable	
Inventory	
Purchasing	
Order Processing	
Project Controller Standard Edition	

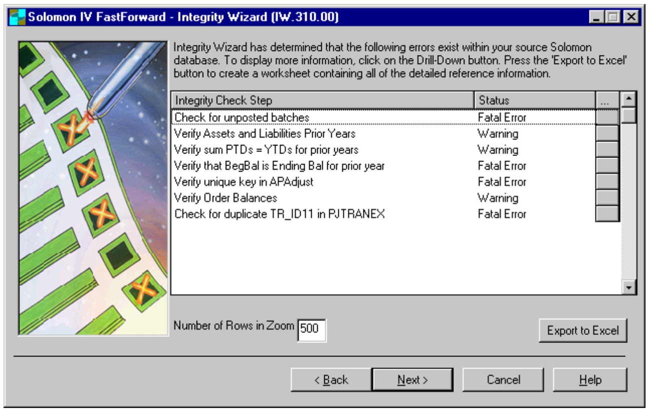
Database Scan: Create Summaries for GL

Current: 00 hrs 00 min 05 sec

Elapsed: 00 hrs 00 min 27 sec

Figure 22: Integrity Wizard (IW.210.00)

22. Any integrity checks that identify errors and cannot be systematically repaired by *Integrity Wizard* will display.



Integrity Wizard has determined that the following errors exist within your source Solomon database. To display more information, click on the Drill-Down button. Press the 'Export to Excel' button to create a worksheet containing all of the detailed reference information.

Integrity Check Step	Status	...
Check for unposted batches	Fatal Error	
Verify Assets and Liabilities Prior Years	Warning	
Verify sum PTDs = YTDs for prior years	Warning	
Verify that BegBal is Ending Bal for prior year	Fatal Error	
Verify unique key in APAdjust	Fatal Error	
Verify Order Balances	Warning	
Check for duplicate TR_ID11 in PJTRANEX	Fatal Error	

Number of Rows in Zoom 500

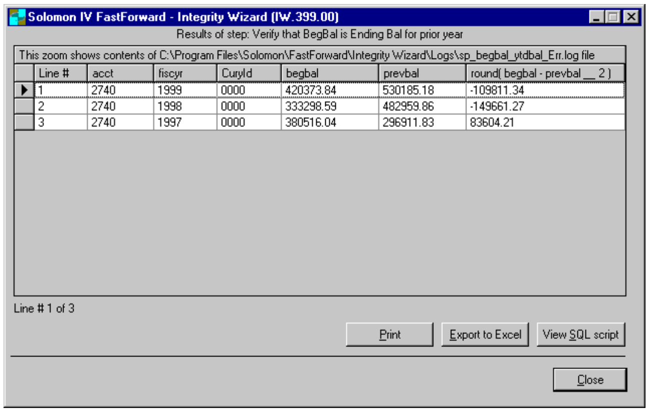
Figure 23: Integrity Wizard (IW.310.00) — Database analysis results

- All queries that identify error conditions will display in the **Integrity Check Step** column. Double-clicking on a step will open a results screen displaying key fields for all records with error conditions for the specific query.
- The **Status** column indicates the severity of the irregular condition. Fatal Error indicates conditions that must be resolved before the database can be migrated to Solomon IV v6.0. Warning indicates conditions that should be reviewed and resolved but will not prevent the migration from continuing.
- The value in **Number of Rows in Zoom** controls the maximum number of records that will display by double-clicking on the **Integrity Check Step** listed. The default value is 500 but may be modified.

- Click **Export to Excel** to export all records with error conditions to Microsoft Excel. Within the Excel spreadsheet, *Integrity Wizard* will create a separate worksheet for each **Integrity Check Step**. Each worksheet contains the records identified as having the error conditions by the Integrity Check. This Excel spreadsheet is a useful tool for analyzing the problematic records and determining the corrective action needed. It also provides an audit trail of the database issues. This export may take a few minutes. Do NOT click on the spreadsheet while records are being written to it. (Note that the active status of the export may only be detectable by the blinking or flashing of the mouse pointer in Excel.) Also, do **NOT** close Excel once it has been opened by *Integrity Wizard* or an error will occur when you attempt to export other data to Microsoft Excel later in *Integrity Wizard*. You will be prompted to save the spreadsheet when you are closing Integrity Wizard after it has finished processing.

Note: The recommended process is to continue with the analysis even if errors are detected that *Integrity Wizard* cannot repair. After the analysis has completed and you have attempted to resolve reported error conditions, rerun *Integrity Wizard* to confirm that all issues were resolved. As mentioned earlier, it is important to reload **Process Flow** when *Integrity Wizard* is rerun against the repaired database.

23. Double-click on an **Integrity Check Step** or click the **Drill-Down** button beside the **Status** column in *Integrity Wizard* (IW.310.00) to display the results set for the selected query.



Results of step: Verify that BegBal is Ending Bal for prior year

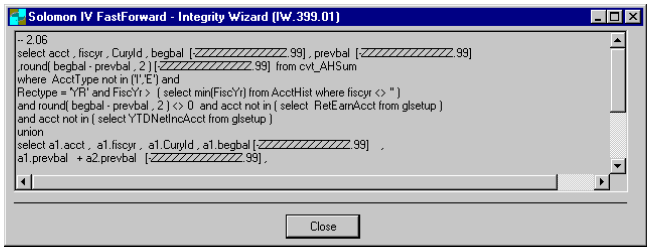
This zoom shows contents of C:\Program Files\Solomon\FastForward\Integrity Wizard\Logs\sp_begbal_ytdbal_Err.log file

Line #	acct	fiscyr	CuryId	begbal	prevbal	round(begbal - prevbal __ 2)
1	2740	1999	0000	420373.84	530185.18	-109811.34
2	2740	1998	0000	333298.59	482959.86	-149661.27
3	2740	1997	0000	380516.04	296911.83	83604.21

Line # 1 of 3

Figure 24: Integrity Wizard (IW.399.00) — Example: Database result set

- The name and path of the log file created for the result set is displayed in the heading. Click **Print** to produce a printed copy of the file.
- The **Export to Excel** button will be disabled if this option was utilized in the prior screen to export all records. If the option was not used in the prior screen, then the records displayed may be exported. If additional results are exported for other queries, they will be added as worksheets to the same spreadsheet.
- Click **View SQL script** to display the SQL script used to return the results. This is for information only. The script may not be modified.

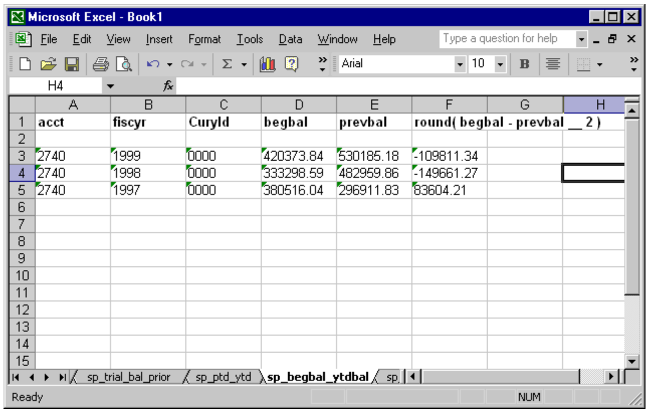


```
-- 2.06
select acct , fiscyr , CuryId , begbal  [-ZZZZZZZZZZZZZ.99] , prevbal  [-ZZZZZZZZZZZZZ.99]
,round( begbal - prevbal , 2 ) [-ZZZZZZZZZZZZZ.99]  from cvt_AHSum
where  AcctType not in ('I','E') and
Rectype = 'YR' and FiscYr >  ( select min(FiscYr) from AcctHist where fiscyr <> '' )
and round( begbal - prevbal , 2 ) <> 0  and acct not in ( select  RetEarnAcct from glsetup )
and acct not in ( select YTDNetIncAcct from glsetup )
union
select a1.acct ,  a1.fiscyr ,  a1.CuryId , a1.begbal [-ZZZZZZZZZZZZZ.99]    ,
a1.prevbal   + a2.prevbal   [-ZZZZZZZZZZZZZ.99] ,
```

Figure 25: Integrity Wizard (IW.399.01) — View SQL script

24. Excel spreadsheets containing the records with the irregular conditions flagged by Integrity Wizard can be created in two different ways.

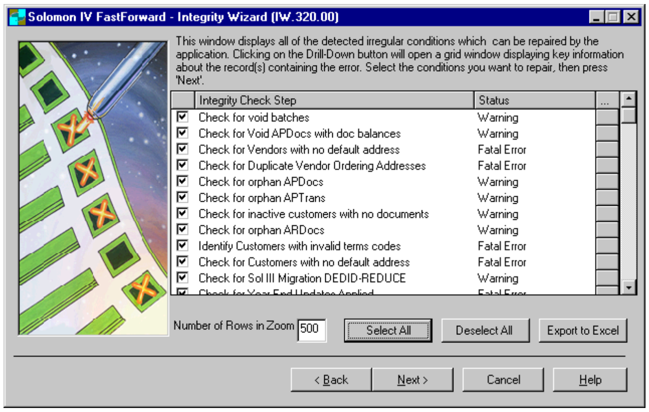
    - Clicking **Export to Excel** in the screen that displays summary level results, *Integrity Wizard* (IW.310.00), creates a spreadsheet workbook that includes the results from all integrity checks showing Fatal Errors or Warnings. Each sheet within the workbook is labeled with the file name of the integrity check script that has returned the error.
    - An alternative method of exporting the results to Excel is to create the spreadsheet one-by-one from the detail results screen for each integrity check displayed by *Integrity Wizard* (IW.399.00). This approach is similar to the previous method; however, it exports results and creates only one worksheet at a time. Additional sheets may be added to the workbook by opening the results of the error and clicking the **Export to Excel** button. The new sheet is added to the existing workbook with the file name of the integrity check script.



acct	fiscyr	CuryId	begbal	prevbal	round(begbal - prevbal __ 2)
2740	1999	0000	420373.84	530185.18	-109811.34
2740	1998	0000	333298.59	482959.86	-149661.27
2740	1997	0000	380516.04	296911.83	83604.21

Figure 26: Microsoft Excel Spreadsheet — Results set, specific integrity checks

25. *Integrity Wizard* has the ability to systematically repair some of the irregular database conditions that could negatively affect the migrated database. These irregular conditions, displayed in the screen below, are distinct from the Fatal Errors and Warnings displayed earlier, for which there are no automatic fixes.



Solomon IV FastForward - Integrity Wizard (IW.320.00)

This window displays all of the detected irregular conditions which can be repaired by the application. Clicking on the Drill-Down button will open a grid window displaying key information about the record(s) containing the error. Select the conditions you want to repair, then press 'Next'.

	Integrity Check Step	Status	...
☑	Check for void batches	Warning	
☑	Check for Void APDocs with doc balances	Warning	
☑	Check for Vendors with no default address	Fatal Error	
☑	Check for Duplicate Vendor Ordering Addresses	Fatal Error	
☑	Check for orphan APDocs	Warning	
☑	Check for orphan APTrans	Warning	
☑	Check for inactive customers with no documents	Warning	
☑	Check for orphan ARDocs	Warning	
☑	Identify Customers with invalid terms codes	Fatal Error	
☑	Check for Customers with no default address	Fatal Error	
☑	Check for Sol III Migration DEDID-REDUCE	Warning	
☑	Check for Year End Updates Applied	Fatal Error	

Number of Rows in Zoom 500

Figure 27: Integrity Wizard (IW.320.00) — Example list of repairable error conditions

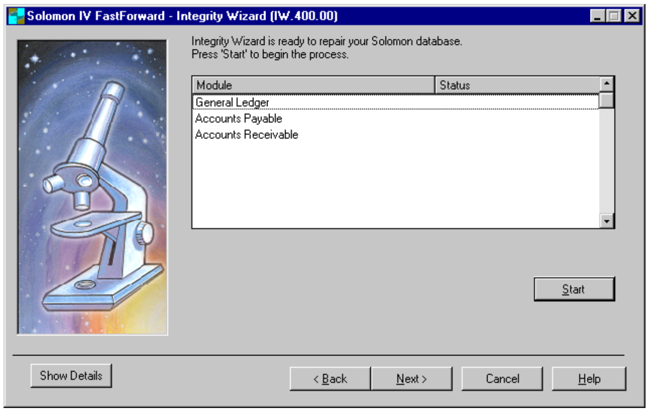
- The results of each **Integrity Check Step** can be reviewed by either double-clicking on the **Integrity Check Step** or clicking the **Drill-Down** button beside the **Status** column. The *Results* window that is launched functions identically to the *Results* window for conditions that could not be systematically repaired.

- Click **Export to Excel** to create a spreadsheet listing all records with irregular conditions. The spreadsheet is not needed to analyze and manually resolve records. However, saving both a printed and electronic copy is recommended for reference purposes.
- Click **Select All** to have *Integrity Wizard* systematically repair all irregular conditions. Selections may also be made individually by clicking in the checkbox beside the desired **Integrity Check Step**.
- Click **Deselect All** to clear all selections. Individual **Integrity Check Step**s may be deselected by clicking in the checkbox beside the desired **Integrity Check Step**.

Once all selections are made, click **Next** to proceed.

Note: Before selecting any automatic repairs to be executed against the database, please review the issue and resolution in “Integrity Check Information” on page 131.

26. *Integrity Wizard* is now ready to begin the database repair process. Click **Start** to launch database repair.



Solomon IV FastForward - Integrity Wizard (IW.400.00)

Integrity Wizard is ready to repair your Solomon database.
Press 'Start' to begin the process.

Module	Status
General Ledger	
Accounts Payable	
Accounts Receivable	

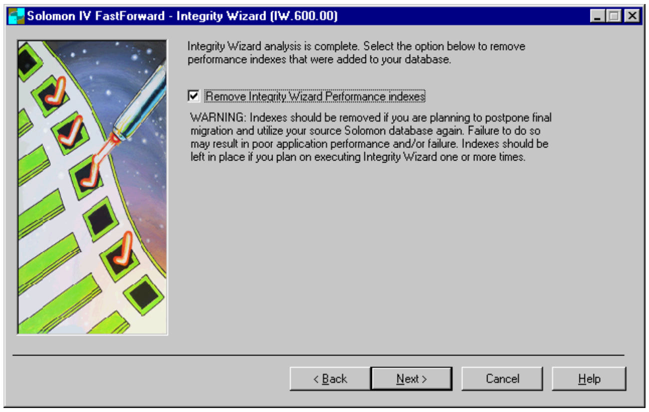
Figure 28: Integrity Wizard (IW.400.00) — Systematic database repair

- Click **Stop** to terminate the repair process. Once stopped, returning to the previous screen, then proceeding to the database repair screen and clicking **Start** resumes the repair process. Records will begin updating from the point where the process was stopped.
- Note that terminating the repair process does not reverse the database transactions that were completed prior to stopping the process.

27. After the database repair process completes, the option to **Remove Integrity Wizard Performance indexes** displays.

- Indexes should be removed if the final migration will be postponed and the database will be used for additional processing prior to the next integrity check.
- Failure to remove indexes may result in poor application performance or failure.

Select the appropriate option, then click **Next** to continue.



Solomon IV FastForward - Integrity Wizard (IW.600.00)

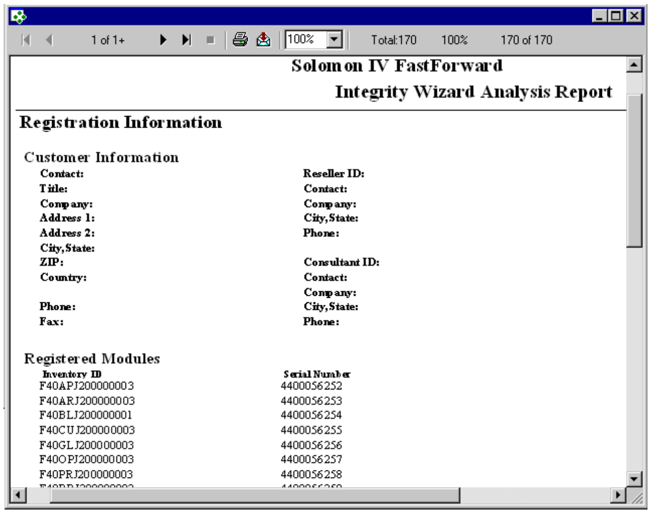
Integrity Wizard analysis is complete. Select the option below to remove performance indexes that were added to your database.

☑ Remove Integrity Wizard Performance indexes

WARNING: Indexes should be removed if you are planning to postpone final migration and utilize your source Solomon database again. Failure to do so may result in poor application performance and/or failure. Indexes should be left in place if you plan on executing Integrity Wizard one or more times.

Figure 29: Integrity Wizard (IW.600.00) — Remove Integrity Wizard Performance indexes

28. The final step in *Integrity Wizard* is to produce the *Integrity Wizard Analysis Report*. Click **Print** to create the report. See “Integrity Wizard Analysis Report” on page 39 for details on the report.



Solomon IV FastForward

Integrity Wizard Analysis Report

Registration Information

Customer Information

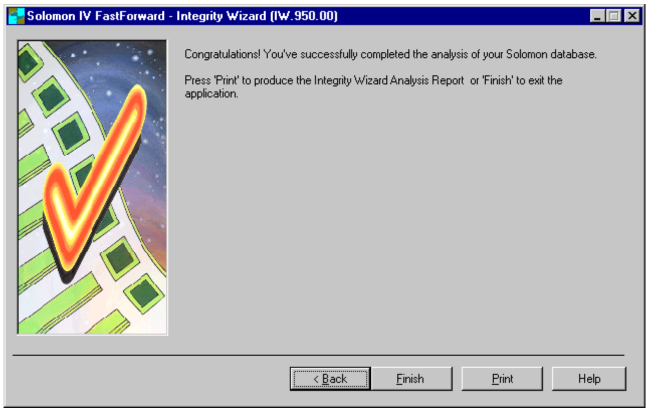
Contact:	Reseller ID:
Title:	Contact:
Company:	Company:
Address 1:	City,State:
Address 2:	Phone:
City,State:	
ZIP:	Consultant ID:
Country:	Contact:
	Company:
Phone:	City,State:
Fax:	Phone:

Registered Modules

Inventory ID	Serial Number
F40APJ200000003	4400056252
F40ARJ200000003	4400056253
F40BLJ200000001	4400056254
F40CUJ200000003	4400056255
F40GLJ200000003	4400056256
F40OPJ200000003	4400056257
F40PRJ200000003	4400056258

Figure 30: Integrity Wizard Analysis Report

29. After printing the *Integrity Wizard Analysis Report*, the first pass with *Integrity Wizard* is complete. If there are no Fatal Errors for any of the **Integrity Check Steps**, the migration process may proceed with FastForward *Transformation Wizard*. Any “unique key” violation errors, those found by checks ending with “_uq,” must be resolved before proceeding. A Preliminary or Test migration may proceed with other unresolved Fatal Errors. However, if the migration is Final Production, or Live, all Fatal Errors identified must be resolved and *Integrity Wizard* must be rerun. *Integrity Wizard* must complete a pass with no Fatal Errors before a Live database migration can proceed on to the Transformation Wizard phase.



Congratulations! You've successfully completed the analysis of your Solomon database.

Press 'Print' to produce the Integrity Wizard Analysis Report or 'Finish' to exit the application.

Figure 31: Integrity Wizard (IW.950.00) — Completing Integrity Wizard

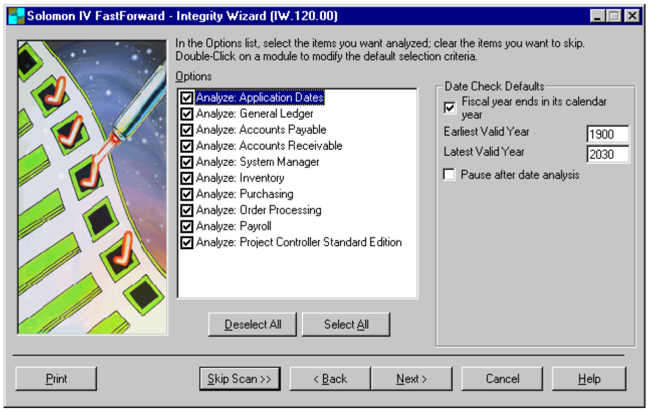
Integrity Wizard Defaults and Module Specifications

Overview

Several of the modules require default values to be defined in *Integrity Wizard* in order for the migration to be completed. This section lists the definitions for the default values within each module. In addition, this section provides additional information and instructions for the migration process for the Distribution Series. The number of new features and enhancements in the Distribution Series and Service Series results in a more complex migration process and requires some post-migration tasks not required by other modules.

Application Date Defaults

Application Date Defaults are used by *Integrity Wizard* to check the database for invalid dates in the database. The SQL queries used will look for any date outside of the year range specified. This check was implemented primarily for use with the Scalable SQL databases in which there may be instances of invalid characters as well as invalid dates such as 2/30/56. The primary use of this function with a SQL Server database is to simply identify and correct dates that may have been erroneously entered by users, such as 3/27/2025.



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Date Check Defaults

☑ Fiscal year ends in its calendar year

Earliest Valid Year 1900

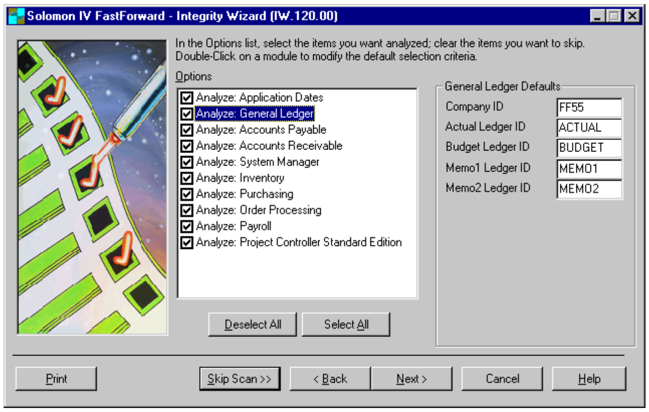
Latest Valid Year 2030

□ Pause after date analysis

Figure 32: Integrity Wizard (IW.120.00) — Application Date Check Defaults

- **Fiscal year ends in its calendar year** is only relevant if the fiscal year is different than the calendar year. For example, if the fiscal year runs June 1999 – May 2000, and the current fiscal year is 1999, then this option should not be checked. However, if the current fiscal year were 2000, then the option would be selected.
- The default for **Earliest Valid Year** is 1900. *Integrity Wizard* will not accept earlier dates. Any date in the database that is earlier than the year entered here will be considered an error.
- The default for **Latest Valid Year** is 2030. *Integrity Wizard* will not accept later dates. Any date in the database that is later than the year entered here will be considered an error.
- Select the option, **Pause after date analysis** if you want to pause *Integrity Wizard* prior to proceeding with the remaining analyses.

General Ledger Defaults



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip.
Double-Click on a module to modify the default selection criteria.

Options

☑ Analyze: Application Dates
☑ Analyze: General Ledger
☑ Analyze: Accounts Payable
☑ Analyze: Accounts Receivable
☑ Analyze: System Manager
☑ Analyze: Inventory
☑ Analyze: Purchasing
☑ Analyze: Order Processing
☑ Analyze: Payroll
☑ Analyze: Project Controller Standard Edition

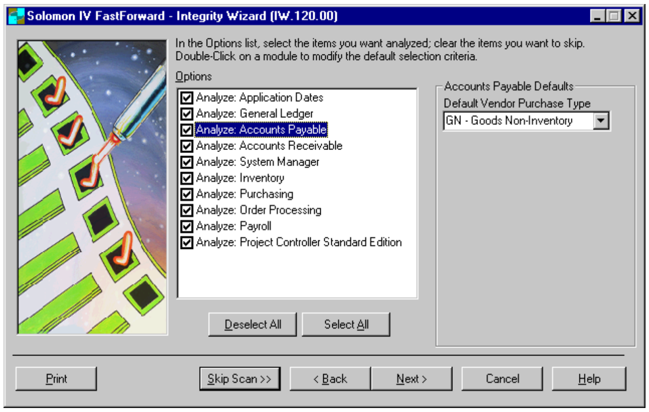
General Ledger Defaults

Company ID	FF55
Actual Ledger ID	ACTUAL
Budget Ledger ID	BUDGET
Memo1 Ledger ID	MEMO1
Memo2 Ledger ID	MEMO2

Figure 33: Integrity Wizard (IW.120.00) — General Ledger Defaults

- Set the **Company ID** to correspond with the desired company ID for the migrated database. A company ID is required. The screen will open with the default value of 0000. The default should be modified to correspond with the client’s specified company ID.
- Complete the **Ledger ID** defaults. Ledger IDs are required. The screen will open with the default values of “ACTUAL”, “BUDGET”, “MEMO1” and “MEMO2” for those ledgers respectively. The defaults should be modified to correspond with the client’s specified Ledger IDs. In Solomon IV v6.0, the **Memo1** and **Memo2** ledgers will be created as BUDGET ledger types and not STATISTICAL ledgers.

Accounts Payable Default



In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Accounts Payable Defaults

Default Vendor Purchase Type: GN - Goods Non-Inventory

Figure 34: Integrity Wizard (IW.120.00) — Accounts Payable Default

- Solomon IV v6.0 requires that each vendor be assigned a default purchase type. The default entered here will be used for all vendors migrated. Select the **Default Vendor Purchase Type** from the drop-down list. Options are:

  GI — Goods for Inventory

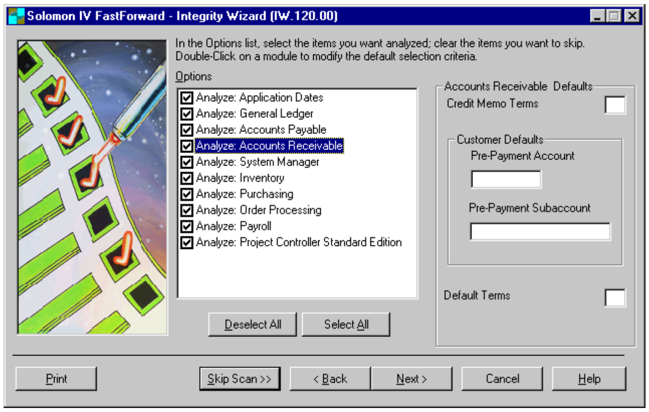
  GN — Goods Non-Inventory

  SE — Services for Expense

  SP — Services for Project

  DL — Description Line

Accounts Receivable Defaults



In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

☑ Analyze: Application Dates
☑ Analyze: General Ledger
☑ Analyze: Accounts Payable
☑ Analyze: Accounts Receivable
☑ Analyze: System Manager
☑ Analyze: Inventory
☑ Analyze: Purchasing
☑ Analyze: Order Processing
☑ Analyze: Payroll
☑ Analyze: Project Controller Standard Edition

Accounts Receivable Defaults

Credit Memo Terms □

Customer Defaults

Pre-Payment Account

Pre-Payment Subaccount

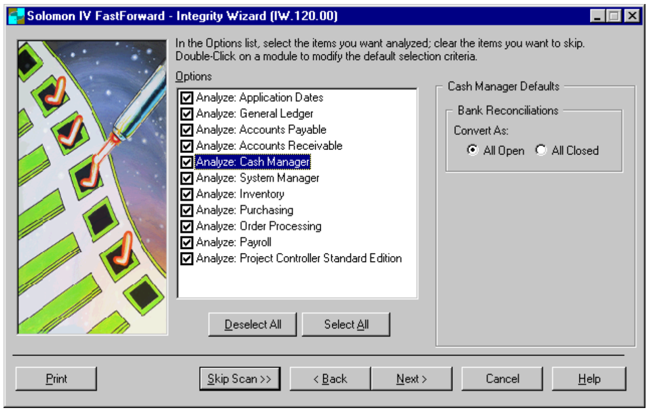
Default Terms □

Figure 35: Integrity Wizard (IW.120.00) — Accounts Receivable Defaults

- Enter a default **Credit Memo Terms** ID. The value must be a valid terms ID in the source database. In Solomon IV 2.06, **Terms ID** was not populated in ARDOC for credit memos. A valid terms ID must be assigned to all credit memos in Solomon IV v6.0 to allow the documents to be included on various Accounts Receivable reports. All credit memos in ARDOC will be assigned this Terms ID.
- Enter a default **Pre-Payment Account**. The pre-payment account must be a valid account in the source database. This account will be assigned as the default pre-payment account for all customers.
- Enter a default **Pre-Payment Subaccount**. The pre-payment subaccount must be established in the source database. This subaccount will be assigned as the default pre-payment subaccount for customers.

- Enter a default value for **Default Terms**. The value must be a valid terms ID in the source database. This value will be used to correct ARDOC and CUSTOMER records that have invalid terms IDs. A valid terms ID is required for all documents and customers in Solomon IV version 6.0. The automatic repair option in *Integrity Wizard* uses this value to correct the records.

Cash Manager Defaults



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip.
Double-Click on a module to modify the default selection criteria.

Options

☑ Analyze: Application Dates
☑ Analyze: General Ledger
☑ Analyze: Accounts Payable
☑ Analyze: Accounts Receivable
☑ Analyze: Cash Manager
☑ Analyze: System Manager
☑ Analyze: Inventory
☑ Analyze: Purchasing
☑ Analyze: Order Processing
☑ Analyze: Payroll
☑ Analyze: Project Controller Standard Edition

Cash Manager Defaults

Bank Reconciliations

Convert As:

● All Open ○ All Closed

Figure 36: Integrity Wizard (IW.120.00) — Cash Manager Defaults

- Bank Reconcilations: The purpose of the Vouchering Stage defaults is to set the ReconcileFlag in the BANKREC table to either Open or Closed. Solomon IV version 6.0 requires that all previous reconciliation records be Closed / Reconciled before new ones are created.

- **All Open** sets migrated BANKREC records to be Open / Not Reconciled. It will be necessary to reconcile these records in Solomon IV version 6.0 prior to creating new reconciliation records.
- **All Closed** sets migrated BANKREC records to be Closed / Reconciled. New reconciliations can immediately be entered in Solomon IV version 6.0.

Payroll Migration Information

Solomon IV version 6.0 Payroll now includes the ability to store month-to-date information on all earnings and deductions data. Since this data was not previously stored in Solomon IV version 2.06, all month-to-date earnings and deduction data in EARNDED will not be populated. However, this data can be initialized in *Earnings Detail* (02.060.01) and *Deduction Detail* (02.060.02) using Initialize mode.

Inventory Defaults and Migration Information

Inventory Migration Information

Due to the number of new Inventory features in Solomon IV v6.0, manual post-migration updates are required for several screens. In addition, clients may require functionality that is not defined in the migration process, but available through post-migration entry. Review the “Post-Migration Inventory Information” section on page 103 for a listing of screens that require updates and for others that contain new functionality in Solomon IV v6.0 but are not setup during the migration process.

FastForward 6.0 does not perform migration of the existing INTRAN records in the source database. Instead INTRAN is rebuilt from data stored in ITEMCOST, ITEMSITE, LOCATION and LOTSERMST tables based on the valuation method used for each inventory item. For items using LIFO and FIFO, the RcptDate and RcptNbr in ITEMCOST is used to create the appropriate cost layers. For items using Specific Identification, the SPECIFICCOSTID in ITEMCOST is used to create the appropriate cost layers.

In order to preserve the flow of the *Inventory Trial Balance* (10.630.00) report for the current inventory period, the INTRAN records are created as follows:

- Transactions for the current period are created to support the current period net inventory movement as contained in ITEMHIST.
- All other transactions required to support the ITEMCOST or ITEMSITE records are created in the last closed inventory period. Please note that the *Inventory Trial Balance* (10.630.00) report for all periods prior to the current period may NOT be accurate due to the absence of transactions.

Example: If your current period (in INSETUP) is 2000-09 then all INTransactions for period 09 will be summarized appropriately to reflect the actual net change of the product during period 09. In period 08, INTransactions are created to substantiate the ending balance with a summary of all historical activity prior to the current period (period 09). It is important to understand this handling because if the desired result is for the migration to maintain the most recent month's net activity then the database should be migrated prior to actually closing the Inventory module. Conversely, using the above example, if it is not necessary to maintain the net activity for the most recent month (period 09), closing the month prior to migration will cause all inventory transactions to be created in period 09.

Lot and serial numbers are managed differently in Solomon IV v6.0 than in previous versions of Solomon IV. Previously there was no validation for assignment of lot or serial numbers regardless of the lot/serial values in inventory maintenance. For example, an item could be set to not use lot or serial numbers; however, numbers could still be assigned during transaction entry. In addition, the assignment could be set as When Issued for an item, however, lot/serial numbers could still be assigned when received. The validation is much tighter in Solomon IV v6.0 and therefore careful review of the existing lot/serial options for each inventory item in the source database is recommended prior to migration.

Transformation Wizard will only migrate lot/serial detail for items where the item has been indicated as either lot or serial numbered and the serial number assignment is set to When Received in *Inventory Maintenance Defaults* (10.250.01). Therefore, if lot/serial numbers were assigned during transaction entry for inventory items not meeting these criteria, the lot/serial number detail will not be migrated for the transaction.

Post-Migration Inventory Information

Inventory contains several new features that are not addressed during the migration process. In some scenarios, the screen is populated but contains new features or functionality not addressed with the migration. In other cases, the screen is not populated by the migration but functionality is available that may need to be implemented to meet client requirements.

The following is a list of screens that are populated but contain additional fields and functionality not populated by the migration. Each of these screens should be opened after migration and reviewed for incomplete data:

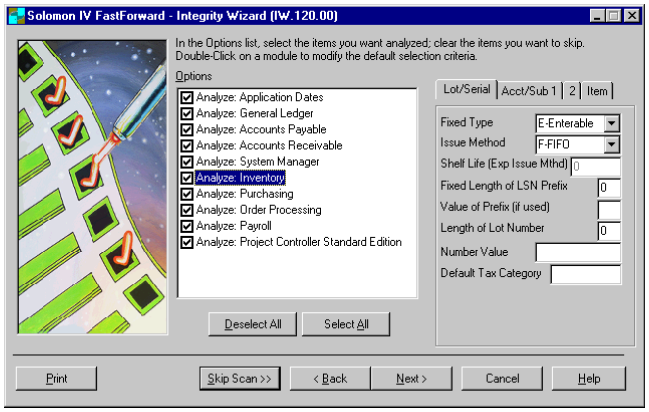
- *IN Setup* (10.950.00)
- *Unit Conversions* (10.270.00
- *Inventory Items* (10.250.00)
- *Product Classes* (10.280.00)
- *Warehouse Bin Locations* (10.340.00)
- *ABC Codes* (10.381.00)

The following screens did not exist in previous versions and are not populated by the migration process:

- *Attribute Definitions* (10.282.00)
- *Item Cross References* (10.380.00)

- *Movement Classes* (10.382.00)
- *Physical Cycles* (10.393.00)
- *Physical Attributes* (10.285.00)
- *Product Lines* (10.286.00)
- *Product Managers* (10.287.00)
- *Reason Codes* (10.350.00)

Inventory Lot/Serial Defaults



In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

☑ Analyze: Application Dates
☑ Analyze: General Ledger
☑ Analyze: Accounts Payable
☑ Analyze: Accounts Receivable
☑ Analyze: System Manager
☑ Analyze: Inventory
☑ Analyze: Purchasing
☑ Analyze: Order Processing
☑ Analyze: Payroll
☑ Analyze: Project Controller Standard Edition

Fixed Type: E-Enterable
Issue Method: F-FIFO
Shelf Life (Exp Issue Mthd): 0
Fixed Length of LSN Prefix: 0
Value of Prefix (if used):
Length of Lot Number: 0
Number Value:
Default Tax Category:

Figure 37: Integrity Wizard (IW.120.00) — Inventory Lot/Serial defaults

If lot and/or serial numbering are not being utilized in the source database, the lot/serial defaults do not need to be populated. However, if the functionality has been utilized in the source database, then the defaults should be completed.

During the migration process, the default values entered will populate *Lot/Serial Number Setup* (10.250.01) for items that were flagged as lot or serial numbered in the source database. Values should be entered that apply to the majority of inventory items since the default values will apply to all items migrated as lot/serial tracked. The default values do not differentiate between lot and serial numbers. Therefore, modification may be required post-migration for specific inventory items.

Important: The lot/serial default values do not populate *IN Setup* (10.950.00). *IN Setup* (10.950.00) should be completed post-migration. In addition, lot/serial defaults for individual inventory items should be reviewed post-migration and modified according to client requirements if necessary.

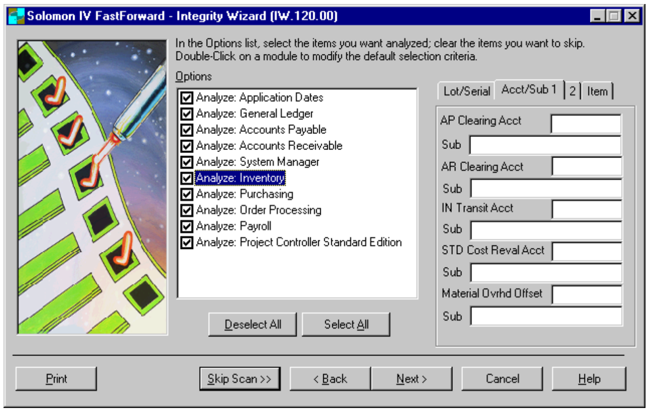
Following are instructions for completing the lot/serial defaults.

Note: The instructions below relate only to completing the defaults on this screen. For a more thorough explanation of how these features are used in Solomon IV v6.0 refer to the Solomon IV v6.0 Inventory User Guide.

1. Select the **Fixed Type** from the drop-down list. **Fixed Type** corresponds with **Prefix Segment Type** in *Lot/Serial Number Setup* (10.250.01). Select one of the following values:
    - E — Enterable; the value can be entered manually and not automatically
    - C — Constant; the value is always the same type
2. Select the **Issue Method** from the drop-down list. **Issue Method** defines the method for issuing the lot/serial numbering assignments. Options include:
    - S — Sequential
    - E — Expiration
    - F — FIFO
    - L — LIFO
3. If the **Issue Method** selected is Expiration, enter the **Shelf Life** as the number of days after receipt into inventory that a lot/serial number for an item will have until its expiration.
4. Specify the **Value of Prefix** using alphanumeric characters to use for the prefix segment.

5. Enter the **Fixed Length of LSN Prefix** to specify the number of characters in the prefix segment of the lot/serial number. The possible values can be between 0 and 12, however the sum of the number of characters in **Fixed Length of LSN Prefix** and the number of characters in the **Length of Lot Number** cannot exceed 25.
6. Enter the **Length of Lot Number** to specify the number of characters in the numeric segment of the lot/serial number.
7. Enter the **Number Value** to be assigned as a lot/serial number. **Number Value** is associated with the numeric segment of the lot/serial number. Specify the next number value to be assigned. The length of the numeric value entered must be equal to the value entered in **Length of Lot Number**.
8. Enter a **Default Tax Category** value that can be used on migrated orders that have tax calculations but no tax category. If the value entered does not currently exist in *Tax Category* (21.280.01) it will be created during the migration. A tax category ID is required on all items that are taxable.

Inventory Account/Subaccount Defaults Tab 1



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip.
Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Figure 38: Integrity Wizard (IW.120.00) — Account/Subaccount defaults — Tab 1

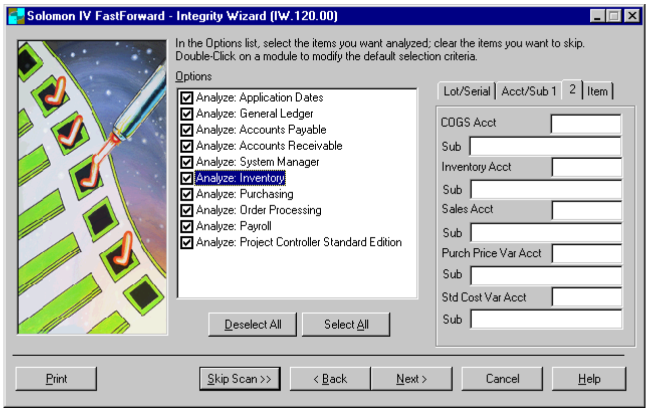
The defaults listed below will populate *IN Setup* (10.950.00). They will not impact migrated inventory items. Inventory items will migrate with their existing account values from the source database. However, if an existing item contains blank values, the values entered here will be used to populate the account/subaccount in the migrated database.

All default values entered must be existing accounts in *Chart of Accounts Maintenance* (01.260.00) for the source database. Since several of the default accounts were not utilized in prior versions, these accounts will need to be added prior to the migration.

- **AP Clearing Acct** specifies the accounts payable clearing account/subaccount that will most often be increased by the receipt of an item.

- **AR Clearing Acct** specifies the accounts receivable clearing account/subaccount that will most often be increased by the sale of an item.
- **IN Transit Acct** specifies the account/subaccount usually affected by inventory transfers.
- **STD Cost Reval Acct** specifies the account/subaccount most often used to assign variance when inventory is revalued.
- **Material Ovrhd Offset** specifies the account/subaccount charged with the material overhead amount (both fixed and variable) from items with standard cost selected as the valuation method.

Inventory Account/Subaccount Defaults Tab 2



In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

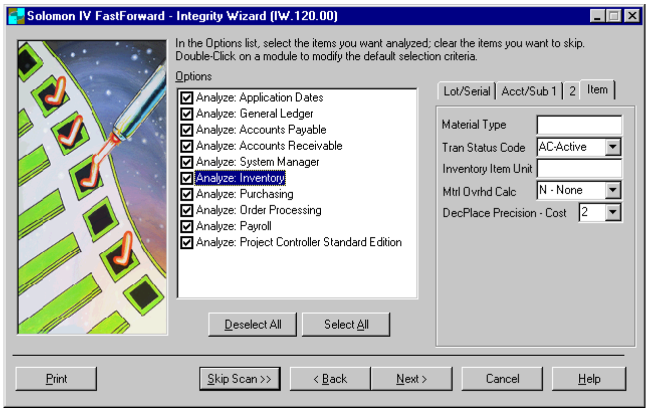
☑ Analyze: Application Dates
☑ Analyze: General Ledger
☑ Analyze: Accounts Payable
☑ Analyze: Accounts Receivable
☑ Analyze: System Manager
☑ Analyze: Inventory
☑ Analyze: Purchasing
☑ Analyze: Order Processing
☑ Analyze: Payroll
☑ Analyze: Project Controller Standard Edition

Figure 39: Integrity Wizard (IW.120.00) — Account/Subaccount defaults — Tab 2

- **COGS Acct/Sub** specifies the cost-of-goods-sold account/subaccount most typically increased by the sale of items.

- **Inventory Acct/Sub** specifies the inventory account/subaccount most typically reduced by the sale of items.
- **Sales Acct/Sub** specifies the sales account/subaccount most typically increased by the sale of items.
- **Purchase Price Var Acct/Sub** specifies the account/subaccount used to assign a variance amount between the purchase price of an item and the standard cost of that item. This is also used for the difference between the receipted amount of an item and the vouchered amount of an item.
- **Std Cost Var Acct/Sub** specifies the account/subaccount most typically used to assign cost variance.

Inventory Item Defaults



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

☑ Analyze: Application Dates
☑ Analyze: General Ledger
☑ Analyze: Accounts Payable
☑ Analyze: Accounts Receivable
☑ Analyze: System Manager
☑ Analyze: Inventory
☑ Analyze: Purchasing
☑ Analyze: Order Processing
☑ Analyze: Payroll
☑ Analyze: Project Controller Standard Edition

Material Type:
Tran Status Code: AC-Active
Inventory Item Unit:
Mtrl Ovrhd Calc: N - None
DecPlace Precision - Cost: 2

Figure 40: Integrity Wizard (IW.120.00) — Inventory Item defaults

The defaults in the **Item** tab populate *IN Setup* (10.950.00) and *Inventory Items* (10.250.00).

- **Material Type** distinguishes broad categories of inventory items. Enter a default value that relates to the majority of inventory items. *Transformation Wizard* will populate *Material Type Maintenance* (21.370.00) with the default value entered as the Material Type Description and ID. Note that the default value should be 10 characters or less in length.
- **Tran Status Code** indicates the default transaction status code that determines how each item may be used in inventory transactions. The value entered as the default will be assigned to ALL inventory items being migrated. Possible values are:
  - AC-Active — The item can be used in all transactions.
  - NP-No Purchase — The item cannot be used in purchasing transactions.
  - NU-No Usage — The item cannot be used in issues or sales transactions.
  - IN-Inactive — The item cannot be used in any transactions.
  - OH-On Hold — The item is temporarily inactive. Transactions can be entered but not processed.
- **Inventory Item Unit** defines the default stocking unit of inventory items. Note that unit migrations are handled differently in Solomon IV v6.0 and therefore careful review of *Unit Migrations* (10.270.00) is recommended post-migration.

- **Mtrl Ovrhd Calc** defines when to calculate overhead for standard cost items. Options are as follows:
  - **When Received** will cause the calculation of overhead to occur when material is received. If selected, Inventory will always value standard cost items using total cost (Direct + Fixed Overhead + Variable Overhead).
  - **When Used** will cause the calculation of overhead to occur when the material is issued or sold. If selected, Inventory will always value standard cost items using direct cost only.
  - **None** indicates that material overhead should not be calculated.
- **DecPlace Precision - Cost** sets the number of decimal places for price/cost. Enter a value that defines the number of decimal places to be used for price and cost of inventory items. The quantity decimal precision is automatically set to 3 during the migration process. This is done to facilitate accuracy in reviewing and updating unit migrations in the migrated database. After the migration is completed, the quantity decimal precision may be changed.

Purchasing Defaults and Migration Information

Purchasing Migration Information

The purchasing process is primarily a direct data migration. However, consideration has to be made with respect to handling of receipt status and voucher stage within purchase orders. In previous versions, the purchase order tracked the quantity received against a line item, but did not contain a receipt status or voucher stage. Solomon IV version 6.0 maintains receipt and vouchering information and therefore, default values must be assigned to existing purchase orders. Details regarding the default options are covered in the “Purchasing Defaults” section.

An additional consideration has to be made for situations where accounts payable vouchers were automatically created (vouchered) for receipts but not released prior to migration. *Integrity Wizard* requires that all accounts payable batches be released prior to migrating the database. However, there may be situations where it is not desirable to release voucher batches created from receipts since the invoice for the procured items has not been received from the vendor. In the instances where it is not practical to release these batches prior to migration, the vouchers must be deleted by *Integrity Wizard* prior to performing *Transformation Wizard* and manually re-entered into Solomon IV v6.0 after the migration.

To address this issue and provide a control document to facilitate easy re-entry after migration, *Integrity Wizard* contains an integrity check that searches for any unreleased accounts payable batches with a status of Hold and a reference to a purchase order number. Any identified records can be exported to an Excel spreadsheet produced through *Integrity Wizard*. The spreadsheet should be printed and maintained for control purposes to ensure that each voucher is re-entered in the migrated database. The automatic repair process in *Integrity Wizard* will delete these batches and associated APDOCs and APTRANs from the database. The migration process also resets the receipt to allow it to be selected for vouchering in Accounts Payable's *Voucher and Adjustment Entry* (03.010.00). Voucher re-entry is facilitated through Accounts Payable's *Voucher and Adjustment Entry* (03.010.00) by pulling up the purchase order number and the receipt number from the respective PV (possible values) listings. As long as the purchase order has not been set to Fully Received or Fully Vouchered then the detail line in the voucher should not have to be manually re-entered. Note that detail lines will have to be manually entered for vouchers when the purchase orders were migrated with the option to set the purchase order to Fully Received or Fully Vouchered.

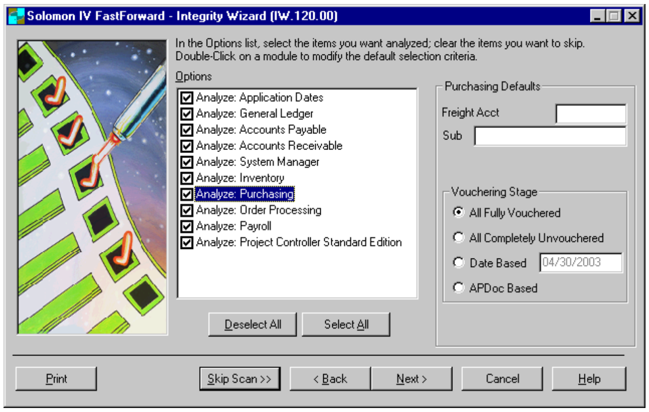
Important: Do not re-enter receipts to create accounts payable vouchers that were deleted. Doing so will cause inventory balances to be misstated. Version 6.0 of FastForward 6.0 includes a feature to renumber line numbers for all purchase orders. Previously, purchase order lines were started at one and incremented by 256. FastForward's process starts line numbers at one and increments them by one for each purchase order number.

Note: If partially vouchered or unvouchered POReceipts exist, after the migration those receipts will need to be manually set to partially vouchered prior to their being vouchered.

Purchasing Post-Migration Information

PO Setup (04.950.00) will need to be updated post-migration to populate additional fields.

Purchasing Defaults


Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip.
Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Purchasing Defaults

Freight Acct

Sub

Vouchering Stage

● All Fully Vouchered

○ All Completely Unvouchered

○ Date Based 04/30/2003

○ APDoc Based

Figure 41: Integrity Wizard (IW.120.00) — Purchasing Defaults

- **Freight Acct/Sub** specifies the expense account/subaccount that will fill in purchase order detail lines for the purchase type of Freight.
- **Vouchering Stage:** The purpose of the Vouchering Stage defaults is to set Receipt Status and Voucher Status values in all existing purchase orders. The options are as follows:

- **All Fully Vouchered** sets all migrated purchase orders to a Closed status, and sets the purchase order as Fully Vouchered and Fully Received. This option will not allow vouchers to be created by utilizing the purchase order number. Additional receipts can only be entered by manually entering detail information. The grid will not populate automatically when the purchase order number is selected in *Receipt/Invoice Entry* (04.010.00).
- **All Completely Unvouchered** sets the purchase order receipt status to Not Received and the voucher status to Not Vouchered. The purchase order status is not altered. This option will allow additional receipts to be entered against existing purchase orders and will also allow new and migrated receipts to be vouchered.

- **Date Based** closes purchase orders based on the cutoff date entered and sets these purchase orders to Fully Vouchered and Fully Received. Purchase orders with a purchase order date after the cutoff date will be set to Not Vouchered and Not Received.
- **APDoc Based** closes purchase orders and sets the purchase order to Fully Vouchered and Fully Received based on the existence of any accounts payable voucher using the purchase order number. This option does have the potential for closing purchase orders that were only partially received.

Order Management Defaults and Migration Information

Order Management Migration Information

Order Management in Solomon IV version 6.0 includes many additional features and functionality not present in prior versions. Therefore the data migration process through FastForward 6.0 should only be considered the foundation for upgrading this module. After the migration is completed, all setup and maintenance screens should be reviewed and updated based on the client’s business requirements. The following information explains how order processing data in Solomon IV 2.06 databases will be migrated to Solomon IV v6.0.

- All **Closed Orders** will be migrated as **Closed Shippers**. Therefore, historical data for closed orders must be viewed through *Shippers* (40.110.00).
- **Standard Orders** will not be migrated due to functionality changes in Solomon IV v6.0.

- Sales **Prices/Discounts** are configured differently in Solomon IV v6.0. In previous versions it was a combination of Invt ID, Price Level, Selling Unit, and Qty break. In Solomon IV v6.0 many more options are available. *Price Levels* (10.290.00) from the Solomon IV 2.06 Inventory module will migrate to *Customer Price Classes* (40.391.00). *Sales Price* (40.380.00) will have **Category** set to Inventory Item and Customer Price Class in order to maintain the same key structure for the migration.
- **Order Types** will migrate as follows:
  - Blanket Order (BO) to Sales Order (SO) — order number will be the version 2.06 order number plus the back order counter
  - Debit Memo (DM) to Debit Memo (DM)
  - Drop Ship (DP) to Sales Order (SO) — with option selected to indicate as Drop Ship
  - Invoice (IN) to Invoice (INVC)
  - Regular Order (OR) to Sales Order (SO)
  - Quote (QT) to Quote (Q)

- Order Management does not assign warehouse locations and lot/serial numbers until the order is shipped. Therefore, warehouse locations and lot/serial numbers will not be migrated on open orders.
- Currently if a tax registration number is entered for a customer in Solomon IV v6.0, no tax will be calculated for any orders entered for the customer in Order Management. Since the functionality differed in Solomon IV 2.06 and open orders with tax are migrated, the CUSTOMER.TAXREGNBR will be migrated into the CUSTOMER.S4FUTURE01 field.

Note: In Solomon IV v6.0, tax is recalculated on orders at different stages as the order passes through the order life cycle.

Order Management Post-Migration Information

FastForward 6.0 now includes a feature to renumber each **LINEREF** field all Order Management sales orders and shippers. Each line starts at one and is incremented by one for each sales order or shipper. The renumbered **LINEREF** is then updated in Accounts Receivable (ARTRAN) to enforce data integrity between the two modules.

Note: In databases that contain many sales orders and/or shippers, this process can take several minutes and possibly hours.

The following screens must be reviewed for new features and updated after the migration to complete Order Management implementation in Solomon IV v6.0.

Order Management:

- *Order Management Setup* (40.950.00)

- *Freight Terms* (40.210.00)
- *Miscellaneous Charges* (40.250.00)
- *Customer Price Classes* (40.391.00)
- *Order Types* (40.200.00)
- *Sales Price* (40.380.00)

Inventory

- *Inventory Items* (10.250.00) — update default inventory IDs added by FastForward
- *Warehouse Bin Locations* (10.340.00)

Shared Information

- *FOB Maintenance* (21.230.00)
- *Ship Via Maintenance* (21.260.00)

Optional Post-Migration Update

Microsoft Business Solutions recommends that the following screens be reviewed and updated in addition to the screens that require post-migration update.

Order Management

- *Certification Text* (40.270.00)
- *Credit Managers* (40.225.00)
- *Customer Contacts* (40.370.00)
- *Inspection* (40.350.00)
- *Item GL Classes* (40.240.00)
- *Lost Sale Codes* (40.280.00)
- *Non-Stock Item* (40.235.00)
- *Payment Types* (40.220.00)
- *User Defaults* (40.340.00)

- *Customer Carriers* (40.807.00)
- *Chain Discounts* (40.330.00)
- *Item Price Classes* (40.390.00)

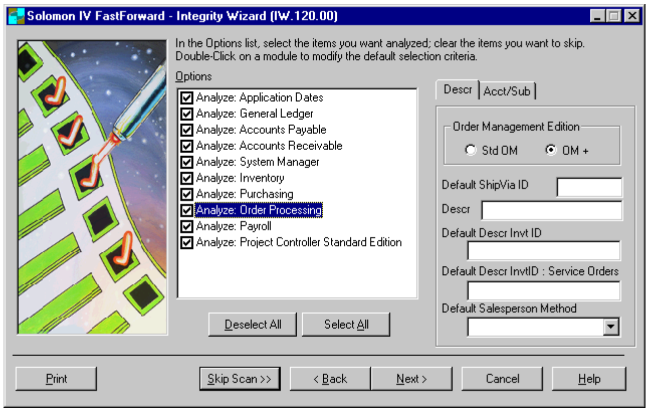
Accounts Receivable

- *Customer Maintenance* (08.260.00) — **Order Management** and **OM GL Account** tabs
- *Shipping Address* (08.262.00)

Shared Information

- *Tax Maintenance* (21.280.00)

Order Management Defaults



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Order Management Edition: ○ Std OM ● OM +

Default ShipVia ID

Descr

Default Descr Invt ID

Default Descr InvtID : Service Orders

Default Salesperson Method

Figure 42: Integrity Wizard (IW.120.00) — Order Management Defaults — Descr tab

Order Management Descr Tab

- Select the appropriate Order Management Edition, **Std OM** or **OM +**. Standard Order Management includes six predefined order types that may not be modified. Order Management Plus offers 17 predefined order types that may be modified and new order types may be added. The selection in this field should be based on what edition the client purchased.
- **Default ShipVia ID** is a required field in all sales orders in Solomon IV v6.0. It is also validated against values established in *Ship Via Maintenance* (21.260.00) in Shared Information. The value entered in this field will be used for all migrated orders that did not previously have a ship via value and will create an entry in *Ship Via Maintenance* (21.260.00) if the value is different from an existing ship via entry in a sales order. *Ship Via Maintenance* (21.260.00) will be populated with all values from sales orders and should be reviewed and updated after the migration to add additional values, modify existing values, or to delete values.
- Enter a description for the **Default ShipVia ID** in **Descr**. This information will be used to create the entry in *Ship Via Maintenance* (21.260.00).

- Enter a **Default Descr Invt ID** that will be used in situations where additional description lines were entered for an order and therefore no inventory ID is associated with the line. Solomon IV version 6.0 requires that each line have an associated inventory ID. The value entered will be used to create a non-stock inventory item that will be used to populate the inventory ID for any lines on an order that do not have inventory ID populated. After the migration this inventory item should be reviewed and modified if necessary to make it unavailable for future use in the migrated database.

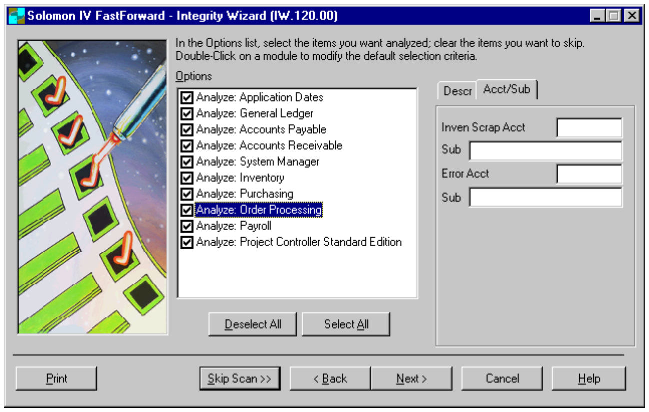
  Important: This default is a required field, it may not be left blank. The value entered may NOT be an inventory ID that already exists in the source database. FastForward 6.0 will create a new inventory item for the value entered.

- Enter an inventory ID in **Default Descr InvtID : Service Orders**. This inventory ID will be used when migrating service orders. If existing values are not valid inventory IDs, this ID will replace the value entered, and the existing data will be moved to S4Future01.

  Important: This default is a required field. It may not be left blank. The value entered must be different than the value entered for **Default Descr InvtID** and the value may NOT be an existing inventory ID in the source database. If duplicate values are entered or values that currently exist in the source database are entered then *Transformation Wizard* will encounter errors when migrating the Inventory and Inventory ADG tables.

- **Default Salesperson Method** uses the method used to select the default salesperson when entering sales orders and shippers. Options are as follows:
  - **C - Customer Ship-to Address** will default the salesperson based on the customer’s ship-to address.
  - **U - User** will default the salesperson based on the information entered for the **Solomon User ID** in *User Defaults* (40.340.00).
  - **B - Both User and Ship-to Address** will check user defaults first and then the customer ship-to address to default the salesperson.
  - **X - No Default** will not default a salesperson.

Order Management Acct/Sub Tab



Solomon IV FastForward - Integrity Wizard (IW.120.00)

In the Options list, select the items you want analyzed; clear the items you want to skip. Double-Click on a module to modify the default selection criteria.

Options

- ☑ Analyze: Application Dates
- ☑ Analyze: General Ledger
- ☑ Analyze: Accounts Payable
- ☑ Analyze: Accounts Receivable
- ☑ Analyze: System Manager
- ☑ Analyze: Inventory
- ☑ Analyze: Purchasing
- ☑ Analyze: Order Processing
- ☑ Analyze: Payroll
- ☑ Analyze: Project Controller Standard Edition

Inven Scrap Acct

Sub

Error Acct

Sub

Figure 43: Integrity Wizard (IW.120.00) — Order Management Acct/Sub tab

The **Acct/Sub** defaults will populate the **Invoicing** tab in *Order Management Setup* (40.950.00). The account and subaccount values entered must be existing accounts in the source database.

- **Inven Scrap Acct** and **Sub** are updated in a transaction when inventory items that are damaged or obsolete are discarded or sold for parts.
- **Error Acct** and **Sub** are used in situations where wildcard combinations yield invalid account numbers when posting invoices.

Service Series Migration Information

Service Series in Solomon IV version 6.0 includes many new features and functionality. Therefore the data migration process through FastForward 6.0 should only be considered the foundation for upgrading Service Series. After the migration is completed, all setup and maintenance screens should be reviewed and updated based on the client’s business requirements. The following information explains how Service Series data in Solomon IV 2.06 databases will be migrated to Solomon IV v6.0.

- All records in XSMFRDETAIL, XSMLBDETAIL, and XSMTMDETAIL will be migrated into a single table, SMSERVDETAIL.
- The Service Call and Service Contract completion and invoicing process in Solomon IV v6.0 will be handled within Service Series, unlike Solomon IV version 2.06 where Order Processing handles the invoicing process.
- In Solomon IV version 2.06 the **On Call Completion COD/Invoice** drop-down list values are COD, Invoice, and Manual. In Solomon IV version 4, the field is labeled **Invoice Type**, and available values are Progress, Invoice, and Manual. COD values will be migrated as Invoice.
- **Invoice Type** in On Call Completion is labeled **Service Type** in Solomon IV v6.0. The option values remain the same (T&M and Flat Rate).
- The Order Number of Service Contract Billing records that had been billed (Status = Processed) are stored as **ARREFNBR** field in Service Series tables in Solomon IV v6.0.

- If Account/Sub Account validation is utilized, all Service Series screens that utilize this functionality will need to be reviewed since the migration process does not do the validation.
- There are several required fields on Service Series Site Maintenance screens that are not required in Solomon IV version 2.06. Therefore records should be reviewed after the migration.
- Account and subaccount values for call types and equipment are defaulted from *AR Setup* (08.950.00).

Integrity Wizard Analysis Report

The *Integrity Wizard Analysis Report* is available for printing at the completion of the *Integrity Wizard* analysis. The report includes information derived from the *Integrity Wizard* analysis as well as information read from the source Solomon IV application subdirectory. The *Integrity Wizard Analysis Report* is intended to assist in the overall analysis of the work effort needed to complete the upgrade process.

In situations where Pervasive.SQL 7 Workstation Edition is being used for the migration the Solomon IV 2.06 application cannot be installed on the same workstation. Therefore, sections of the *Integrity Wizard Analysis Report* that rely on information from the source application directory will not be available.

Following are descriptions of each section in the report:

- **Customer Information** includes information from *Registration* (95.250.00) in the source database, such as customer and reseller name and address.
- **Database Survey** lists each module with record counts from key tables within the modules.
- **Customization Analysis** provides a listing of all screens that have been customized in the source database. Customizations are listed in order of customization level and screen number and include the size of the customization and user IDs for one-user customizations. Customization levels are listed as follows:
  - Language Level — 100
  - Supplemental Product Level — 150
  - All Users — Level 300
  - One User — Levels 400 and 500

Customizations may be moved to the migrated database by exporting the customization from the existing database through Customization Manager, then importing the resulting *.CST files into the migrated database. Because of the substantial functionality and screen changes in Solomon IV v6.0, all customizations should be carefully reviewed prior to importing into the migrated database and thoroughly tested after the import.

- **Templates** lists the screen number, template ID, description, and user ID for templates that were created in the source database. The templates listed should be reviewed with the client to determine if they should be recreated in the Solomon IV v6.0 database. Templates must be manually recreated.
- **Custom Tables** provides the name and record count of any custom tables in the source database. Custom tables will not migrate with the FastForward 6.0 application.
- **Custom Executables** lists *.EXEs from SOL4 subdirectory which are NOT installed by default. In addition, this section lists subdirectories that have been added to normal Solomon IV install.
- **Custom Screens** lists all additions to the SCREEN table.
- **Custom Reports** lists any reports that exist in the SOL4\USR_RPTS subdirectory in the source database. FastForward 6.0 will not migrate custom reports. Any custom reports should be reviewed with the client and recreated in the migrated database.
- **Report Control File Modifications** lists all changes from base level RPTCONTROL table (includes changes to Run Before and Run After EXEs).
- **SOLOMON.INI** lists the settings in the SOLOMON.INI file.

- **Menu Customizations** lists the file name, file date, and size of any *.MNU files in the SOL4 subdirectory with a file name other than DELTA.MNU. In addition, if the size or date/time for DELTA.MNU is different from the original file on the source version CD, then DELTA.MNU will be listed. Menu customizations are not migrated by FastForward 6.0 and must be manually created in the migrated database.
- **Integrity Wizard Summary** lists each module analyzed by *Integrity Wizard*, including the length of time incurred for each to complete integrity checks and the number of Warnings and Fatal Errors encountered. Before running *Transformation Wizard*, all error conditions should be reviewed and appropriate action taken to correct the irregular conditions. Once adjustments are made, *Integrity Wizard* should be run again to ensure that all conditions were successfully repaired. A final live migration can not be run until all Fatal Errors are corrected.
- **Select/Premier Edition Qualifications** indicates whether the source database can be migrated to Select Edition based on *Flexkey Definitions* (21.320.00).

Integrity Check Information

Integrity Check Overview

This section lists the majority of the integrity checks used in *Integrity Wizard* by module. Explanations of each script are included, as well as general recommendations for resolving irregularities.

An indication of Fatal Error under **Explanation** indicates that any errors returned must be resolved before the live migration can be initiated. Results returned from integrity check errors can be viewed and printed in Excel directly through *Integrity Wizard*. These errors do not prevent running a Preliminary Migration through *Transformation Wizard*, however they must be resolved before the live migration is initiated. Databases migrated using the Preliminary Migration functionality will be marked as Preliminary Migrations. Microsoft Business Solutions will not provide technical support for production databases which remain in the Preliminary Migration state.

This information is provided as a guide and is not intended to provide detailed resolutions for database repair issues. All database repairs should be performed by a qualified individual who fully understands the tables, fields and data flows within the Solomon IV application.

Important: Always create a restorable backup of the database prior to performing any database repairs.

System Manager Integrity Checks

1. **SP_ACRIGHTS_UQ** — Verify unique key in ACCESSRIGHTS

   Explanation: Fatal Error — Checks for multiple incidents of the same user ID and type in the ACCESSRIGHTS table. In Solomon IV 2.06 databases CPNYID did not exist in the ACCESSRIGHTS table, however in Solomon IV v6.0 CPNYID is a part of the unique key. In Solomon IV 2.06 access rights were assigned by database name or [ALL]. During the migration process, since there is not necessarily a correlation between the Solomon IV 2.06 database name and the Solomon IV v6.0 CPNYID, there is no way to determine the CPNYID for each database listed in the ACCESSRIGHTS table. FastForward 6.0 initially brings all of the data into the Solomon IV v6.0 database just as it existed in Solomon IV 2.06 (including the old Solomon IV 2.06 database names) and during the migration post-process the CPNYID for ALL records is updated to be the CPNYID of the database being migrated. When there are multiple entries in this table for a user ID/type combination (even though the database name is different) this update will fail because of duplicates.

   Resolution: Review the results in the Excel spreadsheet provided through *Integrity Wizard*. Delete any duplicate records from the ACCESSRIGHTS table. In most instances, deleting all records EXCEPT those where Database Name = [ALL] will be the solution. However, it is important to note that if you are using multiple databases within your system database after migration, you will need to re-enter the access rights for each specific database.

2. **SP_ACDETRIGHTS_UQ** — Verify unique key in ACCESSDETRIGHTS

Explanation: Fatal Error — This integrity check is based on the same logic explained in AP_ACRIGHTS_UQ and checks for multiple incidents of the same user ID, type, and screen number in the ACCESSDETRIGHTS table. There is a unique index on these fields in Solomon IV v6.0; therefore, multiple combinations will cause *Transformation Wizard* to error.

Resolution: Review the results in the Excel spreadsheet provided through *Integrity Wizard*. Delete any duplicate records from the ACCESSDETRIGHTS table. In most instances, deleting all records EXCEPT those where Database Name = [ALL] will be the solution. However, it is important to note that if you are using multiple databases within your system database after migration, you will need to re-enter the access rights for each specific database.

General Ledger Integrity Checks

1. **SP_ACCTHIST_BEGBAL** — Check for beginning balances in YTD Net Income and Balance Sheet (Income/Expense) accounts

   Explanation: Reviews the ACCTHIST records for YTD Net Income and Balance Sheet accounts to verify that there is no beginning balance.

   Resolution: Identified account/subaccount combinations should be updated to have no beginning balance using Initialize mode in *Account History* (01.300.00).

2. **SP_ACCTHIST_GLTRAN_BASE** — Compare Base CURY GLTRAN and ACCTHIST totals for prior years

   Explanation: Account/subaccount combinations are analyzed for previous years where the sum of general ledger transactions does not tie to the period to date balance in *Account History* (01.300.00) in the base currency.

Resolution: Irregular conditions do not have to be repaired prior to the live migration. However, repair is recommended since not repairing them will cause each instance to be present on the general ledger integrity check in the migrated database. The repair for this condition assumes that, since it is for the prior year, financials have been published and the ACCTHIST is correct. Therefore, the resolution requires GLTRANs to be created to support the ACCTHIST balance through a journal entry. Unless the correcting journal entry will balance, this will need to be performed in Initialize mode. The journal entry should be created and released, however it must NOT be posted by the Solomon IV posting process (doing so will cause ACCTHIST to be updated). It MUST be posted manually by updating the BATCH record and GLTRAN for the batch using SQL Scope. The SQL statement to update is as follows:

```
Update BATCH set STATUS = 'P' where
MODULE = 'GL' and BATNBR = '<YOUR
BATNBR>'.
```

```
Update GLTRAN set POSTED = 'P' where
MODULE = 'GL' and BATNBR = '<YOUR
BATNBR>'
```

If the GLTRAN is correct and ACCTHIST is incorrect, you should correct ACCTHIST balance through Initialize mode in the *Account History* (01.300.00).

3. **SP_ACCTHIST_GLTRAN_BASE_CRYR** — Compare Base CURY GLTRAN and ACCTHIST totals for current year

Explanation: Fatal Error — Account/subaccount combinations are analyzed for the current fiscal year where the sum of general ledger transactions does not tie to the period to date balance in ACCTHIST in the base currency. This query will not return errors for account balances that were created to initialize the database. These account balances are recognized as account/subaccount combinations that do not have any GLTRAN activity for the period.

Resolution: This condition must be repaired prior to the live migration. The repair must be done manually through SQL Scope or journal entries.

4. **SP_ACCTHIST_GLTRAN_CURY** — Compare CURY GLTRAN and ACCTHIST totals for prior years

   Explanation: Account/subaccount combinations are analyzed for previous fiscal years where the sum of general ledger transactions does not tie to the period to date balance in ACCTHIST for currencies other than the base currency.

   Resolution: Irregular conditions do not have to be repaired prior to the live migration. However, repair is recommended since not repairing them will cause each instance to be present on the general ledger integrity check in the migrated database. The repair must be done manually through SQL Scope or journal entries.

5. **SP_ACCTHIST_GLTRAN_CURY_CRYR** — Compare CURY GLTRAN and ACCTHIST totals for current year

   Explanation: Fatal Error — Account/subaccount combinations are analyzed for the current fiscal year where the sum of general ledger transactions does not tie to the period to date balance in *Account History* (01.300.00) for currencies other than the base currency.

   Resolution: This condition must be repaired prior to the live migration. The repair must be done manually through SQL Scope or journal entries.

6. **SP_BEGBAL_YTDBAL** — Verify that BEGBAL is ending BAL for prior year

Explanation: Fatal Error — Verifies that the prior year ending balance on balance sheet accounts equals the beginning balance for the accounts in the following fiscal year. Account balances are calculated at an aggregate account level not at the account/subaccount detail level. For example, all ending balances in 1999 for account/subaccount combinations for account 1300 will be aggregated and compared to the beginning balance of the sum of all 1300 accounts for fiscal year 2000. The only exception is that the check considers Retained Earnings and YTD Net Income as the same account. In other words, fiscal year 1999 Retained Earnings + YTD Net Income must equal the fiscal year 2000 beginning balance of retained earnings.

Resolution: Manually resolve through journal entries in the source database. Use the *Trial Balance* (01.610.00) report to verify balances after changes have been made.

7. **SP_CLEAN_VOID** — Check for void batches

   Explanation: Scans records in the BATCH table with a status of V and checks for values in multiple fields that are inconsistent with Void batches. Identified irregularities are not considered fatal errors, and therefore repair is not mandatory in order to continue with the migration.

   Resolution: An automatic repair is available in *Integrity Wizard*. The repair will update specific fields in each record to be consistent with void batches created by the application.

8. **SP_CURYID_CMSETUP** — Verify existence of multiple CURYIDs without CMSetup table

   Explanation: Returns all CURYIDs in the CURRNCY table that are not defined in GLSETUP when the Currency Manager module is not set up. This scenario enables the currency selection field in screens despite the absence of a CMSETUP record.

   Resolution: *Integrity Wizard* includes an automatic repair to remove any records from the CURRNCY table when the CURYID is not defined in the GLSETUP table.

9. **SP_DEL_NLCURY_ACCTHIST** — Check for CURYIDs in ACCTHIST

   Explanation: Fatal Error — Checks for records in ACCTHIST records without a CURYID.

   Resolution: *Integrity Wizard* includes an automatic repair for this error which removes any records from ACCTHIST where the CURYID is null or empty.

   Caution: This repair can only be used if the database uses a single currency. Solomon creates a duplicate ACCTHIST record with correct base currency during the posting process.

10. **SP_GLTRAN_EDITSCRNNBR** — Verify that BATCH.EDITSCRNNBR is not blank for general ledger batches

    Explanation: Returns all BATNBRs from the BATCH table where the **EDITSCRNNBR** field is null or empty.

    Resolution: *Integrity Wizard* includes an automatic repair to update the EDITSCRNNBR to 01010 (*Journal Transactions* (01.010.00)) where it was previously null or empty.

11. **SP_GLTRAN_POSTED_STS** — Verify that GLTRAN.POSTED matches BATCH.STATUS

    Explanation: Returns all records in the GLTRAN table where posted status does not match the BATCH status in the BATCH table.

    Resolution: *Integrity Wizard* includes an automatic repair to update GLTRAN.POSTED to match BATCH.STATUS.

12. **SP_INVAL_ACCT_GLTRAN** — Check for invalid accounts in GLTRAN

    Explanation: Identifies all GLTRAN records with a value in the ACCT field that is not a valid account in the Chart of Accounts list.

    Resolution: Identified records should be researched further. Once researched and verified, either the correct ACCT value should be updated in the GLTRAN table or the record should be deleted.

13. **SP_INVAL_CURYID_ACCT** — Check for invalid currency IDs in ACCOUNT

    Explanation: Identifies all ACCOUNT records with a value in the CURYID field that is not a valid currency ID.

    Resolution: *Integrity Wizard* includes an automatic repair to blank out the CURYID value in the ACCOUNT table if CURYID is not a valid value.

14. **SP_INVAL_SUB_GLTRAN** — Check for invalid subaccounts in GLTRAN

    Explanation: Identifies all GLTRAN records with a value in the SUB field that is not a valid subaccount in the Subaccount list.

    Resolution: Identified records should be researched further. Once researched and verified, either the correct SUB value should be updated in the GLTRAN table or the record should be deleted.

15. **SP_MANUAL_BATCH** — Check for manual batches

    Explanation: Identifies all general ledger batches with a BATTYPE of M in the BATCH table. The check is included primarily for databases being migrated from Solomon III where manual batches are handled differently than in Solomon IV.

    Resolution: If the migration originated as a Solomon III database, manual batches with a status of P should be set to be Non-Recurring. All other Solomon III manual batches should be deleted and manually re-entered in the migrated database if appropriate.

    Solomon IV manual batches may remain in the database, however it is recommended that the client review and delete all manual batches if they are no longer being used.

An automatic repair is available in *Integrity Wizard*, however it should be used with caution as it deletes all general ledger batches with a BATTYPE of M. To delete specific batches instead of all batches, use *Journal Transactions* (01.010.00).

16. **SP_NLCURY_GLTRAN** — Check for non-repairable CURYIDs in GLTRAN

    Explanation: Fatal Error — Returns records if the database is multi-currency and if there are GLTRAN records without a CURYID. This condition should never exist in the database.

    Resolution: Because programmatically it cannot be determined what the correct CURYID should be, the records must manually be repaired.

17. **SP_NLYR_GLTRAN** — Check for non-repairable FISCYRS in GLTRAN

    Explanation: Fatal Error — Returns GLTRAN records that have no fiscal year and no period-to-post value in the record.

    Resolution: These records will need to be manually repaired by reviewing the associated batch and updating the record with the appropriate fiscal year through SQL Scope.

18. **SP_NODOC_GLBATCH** — Check for general ledger batches with no TRANS

    Explanation: Identifies all general ledger batches that do not have a status of Voided or Deleted, and have no transactions associated with them. Error results are for informational purposes only and records identified will not necessarily have a negative impact in the migrated database.

    Resolution: An automatic repair is available in *Integrity Wizard*. The repair updates identified records by setting the status to Void.

19. **SP_PERPOST_GLTRAN_BATCH** — Compare PERPOST in GLTRAN and BATCH

    Explanation: Fatal Error — Compares the period-to-post in each general ledger batch with the period-to-post in the related general ledger transaction lines and returns any batches were they do not match.

    Resolution: An automatic repair is available in *Integrity Wizard*. The repair updates GLTRANs by setting the period-to-post equal to the corresponding batch record’s period post.

20. **SP_POST_NOREL_BATCH** — Check for posted batches marked as unreleased

    Explanation: Fatal Error — The BATCH table is queried for records where STATUS = P and RLSED = FALSE. These are batches that have been posted but the released flag in the batch record has not been appropriately set to True.

    Resolution: Any batch records returned must be repaired prior to migration by setting BATCH.RLSED to TRUE.

21. **SP_PTD_YTD** — Verify sum PTDs = YTDs for prior years

    Explanation: Identifies all account/subaccount combinations where the previous period's YTDBAL + current PTDBAL <> current period's YTDBAL. This check looks at all records in prior fiscal years.

    Resolution: These conditions do not result in a Fatal Error, therefore repair is optional. This condition likely exists with other GL to ACCTHIST integrity problems. The condition may be repaired using Initialize mode in *Account History* (01.300.00), however changes should be made carefully. Typically financial reports have already been issued for the period(s) affected and any initialization may change the financial results. After correcting the error, rerun the integrity check(s), Create GL Summaries and Compare AcctHist to GLTran, to ensure that your work did not cause other problems. There are four integrity checks that compare ACCTHIST to GLTRAN to account for multi-currency environments and current as well as prior year totals.

22. **SP_PTD_YTD_CRYR** — Verify sum PTDs = YTDs for current year

    Explanation: Fatal Error — Identifies all account/subaccount combinations where the previous period's YTDBAL + current PTDBAL <> current period's YTDBAL. This check looks at all records in the current fiscal year.

    Resolution: This condition likely exists with other GL to ACCTHIST integrity problems. The condition may be repaired using Initialize mode in *Account History* (01.300.00), however changes should be made carefully. Typically financial reports have already been issued for the period(s) affected and any initialization may change the financial results. After correcting the error, rerun the integrity check(s), Compare AcctHist to GLTran, to ensure that your work did not cause other problems. There are four integrity checks that compare ACCTHIST to GLTRAN to account for multi-currency environments and current as well as prior year totals.

23. **SP_STAT_TRIAL_BAL** — Create Summaries for GL

    Explanation: This statement populates temporary tables that are used for various integrity checks involving general ledger tables, such as determining whether the trial balance is in balance, as well as comparing GLTRAN to ACCTHIST.

    Important: This integrity check MUST be rerun each time the general ledger queries are run after database repairs have been made against results for other queries. Otherwise, *Integrity Wizard* will continue to return errors against the corrected queries.

Resolution: Not applicable, this check will not return an error.

24. **SP_SUSP_BATCH** — Check for suspended batches

    Explanation: Fatal Error — Pre-migration steps include repairing all suspended or partially released batches. This check searches the BATCH table for any batch with a status of S.

    Resolution: Review the results returned in the Excel spreadsheet and manually repair any identified batches in the source database.

25. **SP_TRIAL_BAL_PRIOR** — Verify assets and liabilities in prior fiscal years

    Explanation: Checks for the condition where assets do not equal liabilities and equity in years prior to the current fiscal year.

    Resolution: Irregular conditions do not have to be repaired prior to the live migration. The repair must be done manually through SQL Scope or journal entries.

26. **SP_TRIAL_BAL_CURRENT** — Verify assets and liabilities in the current fiscal year

    Explanation: Fatal Error — Checks for the condition where assets do not equal liabilities and equity in the current fiscal year. Assets MUST equal liabilities and equity for ALL periods in the current fiscal year. A live migration may not proceed until this condition is repaired.

    Resolution: This condition must be manually repaired. Delete history, enter correcting journal entries or modify *Account History* (01.300.00) in Initialize mode to facilitate a balanced trial balance. Prior to any initialization a restorable backup should be created. Be aware that any initialization performed to a balance sheet account will roll forward to all subsequent years. Once repairs are made, rerun the Create GL Summaries integrity check in addition to Verify Assets and Liabilities to confirm that the modifications made corrected the condition.

27. **SP_UNBAL_BATCH** — Check for unbalanced unreleased batches

    Explanation: Fatal Error — Scans the BATCH table for records where RLSED = False and the where the debit and credit totals do not equal the control total, causing an out of balance situation. All batches must be balanced, and released prior to the live migration.

    Resolution: Identified batches must be balanced and released in the source database. In addition, all batches through the current period MUST be posted prior to the live migration.

28. **SP_UNPOST_BATCH** — Check for unposted batches

Explanation: Fatal Error — Checks the BATCH table for any unposted batches with a BATTYPE other than M (Manual) or R (Recurring) where PERPOST is less than or equal to the current period.

Resolution: Identified batches must be posted prior to migrating.

29.**SP_UP_NLCURY_GLTRAN** — Check for repairable CuryIDs in GLTran

Explanation: Checks GLTRAN for records without a valid CURYID that can automatically be repaired by *Integrity Wizard*.

Resolution: *Integrity Wizard* includes an automatic repair for identified records. If the database is multi-currency, the CURYID will be taken from the corresponding BATCH record. If the database is single currency, the CURYID is taken from GLSETUP.

30. **SP_UP_NLYR_GLTRAN** — Check for repairable fiscal years in GLTRAN

    Explanation: Returns GLTRAN records that have no fiscal year in the record but the fiscal year can be determined by the value in the PERPOST.

    Resolution: *Integrity Wizard* includes an automatic repair for identified records. The update sets the fiscal year to the year indicated in PERPOST in the identified records.

31. **SP_UP_YR_GLTRAN** — Verify that FISCYR matches the first four character of PERPOST

    Explanation: Selects records in GLTRAN where FISCYR matches the first four characters of PERPOST.

    Resolution: *Integrity Wizard* includes and automatic repair to update FISCYR to equal the first four characters in PERPOST in GLTRAN.

32. **SP_YTD_NETINCOME** — Verify YTD net income

    Explanation: Verifies that the balances in YTD_NETINCOME are equal to income less expenses for the same period (YTD). The best way to do this is to run a trial balance for the YTD net income account and run another trial balance for income and expenses only. The net of the income and expenses must equal the total balances of the YTD net income account. This error is often accompanied by one or more additional general ledger integrity violations.

Resolution: This condition must be manually resolved through either initialization or journal transaction in the source database. The most common cause of this error is that a journal entry was posted directly to the YTD net income account. In the instance where a journal entry was recorded directly to this account, it most probably should be reversed. To verify the correction run the *Trial Balance* (01.610.00) report for the YTD net income account. Then run the report again for income and expense accounts only. The net of the income and expense must equal the balances in the YTD net income account.

Accounts Payable Integrity Checks

1. **SP_AP_SUM** — Create summaries for AP

   Explanation: This series of statements populates temporary work tables used by *Integrity Wizard* to compare a vendor's current and future balances with open docs. This query never returns any results directly, however it MUST be rerun each time SP_APDOCBAL_VENDBAL is run.

   Resolution: Not applicable since the integrity check will not return errors.

2. **SP_APADJUST_UQ** — Verify unique key in APADJUST

   Explanation: Fatal Error — Creates a temporary table populated with data from APADJUST, then queries the table for duplicate combinations of ADJDREFNBR, ADJDDOCTYPE, ADJGREFNBR, ADJGDOCTYPE, VENDID, ADJGACCT, and ADJGSUB which would violate a unique index in Solomon IV v6.0.

   Resolution: Manual adjustments must be made to the source database to remove any duplicate records.

3. **SP_APDOC_ALPHA_REFNBR** — Checks for alpha characters in REFNBR in APDOC

   Explanation: Searches for alpha characters in APDOC REFNBR. The *Check Register* (20.600.00) report will not print if there is an instance of alpha characters in accounts payable check documents.

Resolution: It is recommended that you test the preliminary migration to see if this is a problem. As of the date of this publication, this issue has been reported as a defect and will be repaired in due course. The options are to leave the reference numbers as is or remove them from the database. The client should be notified of the issue and determine the most plausible approach.

4. **SP_APDOC_STS** — Verify consistency of APDOC statuses and balances

   Explanation: Identifies all accounts payable documents where the document balance is inconsistent with the OPENDOC status. For example, when the document balance is zero, the document should be closed. When the document balance is non-zero, the document status should be open.

   Resolution: *Integrity Wizard* includes an automatic repair that updates all open documents that have a zero balance to a closed status. Documents that are closed but have a non-zero balance will have to be researched and manually repaired if necessary.

5. **SP_APDOC_UQ** — Verify unique key in APDOC

   Explanation: Fatal Error — The following fields combine to form the unique key in APDOC in version 4: **ACCT**, **SUB**, **DOCTYPE**, **REFNBR**, **RECORD ID**. *Integrity Wizard* checks for and returns any records with duplicate combinations of these fields.

   Resolution: Documents must be removed from the database or the unique key must be altered to be unique prior to migration.

   Note: If the key is modified, you must ensure that the key is also changed in related tables. The tables affected depend on which segment of the key is modified.

6. **SP_APDOC_VOID** — Check for void APDOCs with DOC balances other than zero

   Explanation: Identifies all APDOCs with a status of Void that have document balances other than zero. Once a document has been voided, the DOCBAL should be zero.

   Resolution: *Integrity Wizard* includes an automatic repair option that will set the DOCBAL to zero for all voided documents.

7. **SP_APDOCBAL_APADJ** — Verify DOCBAL equals calculated DOCBAL in APDOCS

   Explanation: Fatal Error — Checks open APDOCs to determine if DOCBAL is equal to the calculated document balance based on the associated records in APADJUST.

   Note: This check does NOT validate closed APDocs.

Resolution: Any records returned must be researched and manually adjusted. The APDOC.DOCBAL and CURYDOCBAL should be updated based on ORIGDOCAMT less the APADJUSTMENT total for the document.

8. **SP_APDOCBAL_VENDBAL** — Compare VENDOR current and future balances to APDOC balances

   Explanation: Returns all instances where a vendor's current and/or future balances do not equal the sum of the open document balances for the vendor.

   Resolution: *Integrity Wizard* includes an automatic repair option that will update the vendor's current balance and future balance to the calculated value.

9. **SP_INVAL_ACCT_APDOC** — Check for invalid accounts in APDOC

   Explanation: Fatal Error — Identifies all APDOC records with a value in the ACCT field that is not a valid account in the Chart of Accounts list.

   Resolution: Identified records should be researched further. Once researched and verified, either the correct ACCT value should be updated in the APDOC table or the record should be deleted.

10. **SP_INVAL_ACCT_APTRAN** — Check for invalid accounts in APTRAN

    Explanation: Identifies all APTRAN records with a value in the ACCT field that is not a valid account in the Chart of Accounts list.

    Resolution: Identified records should be researched further. Once researched and verified, either the correct ACCT value should be updated in the APTRAN table or the record should be deleted.

11. **SP_INVAL_SUB_APDOC** — Check for invalid subaccounts in APDOC

    Explanation: Identifies all APDOC records with a value in the SUB field that is not a valid subaccount in the Subaccounts list.

    Resolution: Identified records should be researched further. Once researched and verified, either the correct SUB value should be updated in the APDOC table or the record should be deleted.

12. **SP_INVAL_SUB_APTRAN** — Check for invalid subaccounts in APTRAN

Explanation: Identifies all APTRAN records with a value in the SUB field that is not a valid subaccount in the Subaccounts list.

Resolution: Identified records should be researched further. Once researched and verified, either the correct SUB value should be updated in the APTRAN table or the record should be deleted.

13.**SP_NEG_DOCBAL_APDOC** — Check for documents with a negative balance

Explanation: Fatal Error — Returns all accounts payable documents (open or closed) that have a document balance below zero.

Resolution: Identified records should be researched to determine the correct document balance and status. Once determined, the DOCBAL field in APDOC should be updated. The following SQL statement can be used:

```
"UPDATE APDOC SET DOCBAL = XXXX.XX
WHERE ACCT = 'XXXXXX' AND SUB =
'XXXXXX' AND REFNBR = 'XXXXXX' AND
DOCTYPE = 'XX';"
```

14.**SP_NLBAT_APDOC** — Check for BATNBRs in APDOC

Explanation: Returns all APDOCs with null BATNBRs. This condition should not exist and probably indicates a record that should be removed from the database. All valid documents should have non-null batch numbers.

Resolution: Identified records should be researched further. Once researched and verified that the APDOCs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APDOC is valid, the record must be updated to include the appropriate BATNBR.

15. **SP_NLBAT_APTRAN** — Check for BATNBRs in APTRAN

    Explanation: Returns all APTRANs with null BATNBRs. This condition should not exist and probably indicates a record that should be removed from the database. All valid transactions should have non-null batch number.

    Resolution: Identified transactions should be researched further. Once researched and verified that the APTRANs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APTRAN is valid, the record must be updated to include the appropriate BATNBR.

16. **SP_NLCURY_APDOC** — Check for CURYIDs in APDOC

    Explanation: Returns all APDOCs with null CURYIDs. This condition should not exist and usually indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the APDOCs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APDOC is valid, the record must be updated to include the appropriate CURYID.

17. **SP_NLCURY_APTRAN** — Check for CURYIDs in APTRAN

Explanation: Returns all APTRANs with null CURYIDs. This condition should not exist and usually indicates a record that should be removed from the database.

Resolution: Identified records should be researched further. Once researched and verified that the APTRANs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APTRAN is valid, the record must be updated to include the appropriate CURYID.

18. **SP_NLREF_APDOC** — Check for REFNBRs in APDOC

    Explanation: Returns all APDOCs with null REFNBRs. This condition should not exist and usually indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the APDOCs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APDOC is valid, the record must be updated to include the appropriate REFNBR.

19. **SP_NLREF_APTRAN** — Check for REFNBRs in APTRAN

    Explanation: Returns all APTRANs with null REFNBRs. This condition should not exist and usually indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the APTRANs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APTRAN is valid, the record must be updated to include the appropriate REFNBR.

20. **SP_NLVEND_APDOC** — Check for VENDIDs in APDOC

    Explanation: Returns all APDOCs with null vendor IDs. This condition should not exist and probably indicates a record that should be removed from the database. All valid documents should have a non-null vendor ID.

    Resolution: Identified records should be researched further. Once researched and verified that the APDOCs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APDOC is valid, the record must be updated to include the appropriate VENDID.

21. **SP_NLVEND_APTRAN** — Check for VENDIDs in APTRAN

    Explanation: Returns all APTRANs with null vendor IDs. This condition should not exist and probably indicates a record that should be removed from the database. All valid transactions should have a non-null vendor ID.

    Resolution: Identified records should be researched further. Once researched and verified that the APTRANs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the APTRAN is valid, the record must be updated to include the appropriate VENDID.

22. **SP_NOBATCH_APDOC** — Check for orphan APDOCs

Explanation: Returns all APDOCs that do not have a corresponding batch record. This condition should not exist in the database. The batch record not existing will prohibit the document from being retrieved in *Voucher and Adjustment Entry* (03.010.00) or *Manual Check Entry* (03.030.00) and possibly on some reports. However, the documents will still display in *Document Maintenance* (03.250.00) where batch numbers are not keys to displaying the document.

Resolution: *Integrity Wizard* offers two options to automatically resolve the database condition. The first option is to delete the identified APDOC records. (Note that this option will delete even Open APDOCS.) The second option provides the ability to create BATCH records for the orphaned APDOCs. The options are presented after clicking **Next** in *Integrity Wizard* (IW.320.00).

23. **SP_NODOC_APADJUSTD** — Check for APADJUSTED DOCs without APDOCs

Explanation: Identifies all APADJUST records with an ADJDREFNBR (voucher or credit adjustment) that does not exist in APDOC. Since the original REFNBR does not exist in APDOC, it could potentially be created again with a duplicate number in the migrated database, causing a query and reporting issue. Additionally, if the ADJGREFNBR (check) has not been closed then subsequently voiding this check will cause the void check batch to suspend when the process tries to re-instate the original document paid.

Resolution: It is recommended that if the check has not been cleared that this list be printed and maintained by the person responsible for voiding checks. In the event that this check needs to be subsequently voided, the original voucher document will need to be recreated in the database. Recreating the voucher should be performed by a consultant very experienced in Solomon database repair.

24. **SP_NODOC_APADJUSTG** — Check for APADJUSTING records with APDOCs

    Explanation: This query returns all APADJUST records with an ADJGREFNBR that does not exist in APDOC. Since the adjusting document is not in APDOC, it may be created again with a duplicate number in the migrated database, causing a query and reporting issue.

    Resolution: Document the issue. The probability of this causing an issue in the migrated database is small.

25. **SP_NODOC_APBATCH** — Check for accounts payable batches with no DOCS

    Explanation: Identifies accounts payable batches that have no documents. This condition may be the result of purging data.

    Resolution: Review identified batches and verify that they are invalid. *Integrity Wizard* includes an automatic repair that will set the BATCH status to Void for all identified batch records. This will prohibit the batch record from being selected in the respective data entry screen.

26. **SP_NODOC_APTRAN** — Check for orphan APTRANs

    Explanation: Identifies APTRAN records with no associated APDOC. These records have no use in the database and should be deleted.

Resolution: *Integrity Wizard* will delete all orphan APTRANs with the exception of records with a TRANTYPE of DT and a DRCR of R. Records with these values do not have associated APDOCs.

27. **SP_NULL_ACCTSUB_APCK** — Check for blank account/subaccount in uncleared accounts payable checks

    Explanation: Fatal Error — Identifies uncleared check records in APDOC with blank account or subaccount.

    Resolution: *Integrity Wizard* includes an automatic repair that will update identified records with the appropriate account and subaccount from APSETUP.

28. **SP_PERPOST_APDOC_BATCH** — Compare PERPOST in APDOC and BATCH

    Explanation: Fatal Error — Identifies APDOC records where the PERPOST is not equal to the associated BATCH PERPOST. BATCH PERPOST determines the period in which the document was posted in General Ledger.

    Resolution: *Integrity Wizard* includes an automatic repair that will update the APDOC PERPOST to equal the associated BATCH PERPOST.

29. **SP_PERPOST_APTRAN_BATCH** — Compare PerPost in APTRAN and batch

    Explanation: Fatal Error — Identifies APTRAN records where the PERPOST is not equal to the associated BATCH PERPOST. BATCH PERPOST determines the period in which the transactions were posted in General Ledger. This condition will cause a reporting problem in the *Detail General Ledger* (01.620.00) report and possibly others within the Account Payable module.

    Resolution: *Integrity Wizard* includes an automatic repair that will update the APTRAN PERPOST to equal the associated BATCH PERPOST.

30. **SP_POADDRESS_UQ** — Check for duplicate Vendor Ordering Addresses

    Explanation: Fatal Error — Identifies records in the POADDRESS table with the same VENDID/ORDFROMID combination. This situation is not permitted in Solomon IV version 6.0.

    Resolution: *Integrity Wizard* includes an automatic repair that will delete the duplicate Vendor Ordering Address records.

31. **SP_UNCLEARED_APCK** — Check for uncleared checks with accounts payable adjustments

    Explanation: Fatal Error — Identifies APDOC records with a DOCTYPE of CK or HC where the PERCLOSED is null (or empty) and associated APTRANs do not exist for the document.

Resolution: This will cause a problem if the check subsequently is voided. Accounts payable transactions must be recreated for this document or the document must be deleted from the database (not recommended). Adding appropriate accounts payable transactions should be performed by an experienced Solomon database repair consultant.

32.**SP_VEND_NO_POADDR** — Check for vendors with no default address

Explanation: Fatal Error — Returns all vendors with no default order address defined in POADDRESS.

Resolution: *Integrity Wizard* contains an automatic repair to create a POADDRESS record for each vendor ID returned using the address information that is contained in the VENDOR table.

Accounts Receivable Integrity Checks

1. **SP_AR_SUM** — Create summaries for Accounts Receivable

   Explanation: This series of statements populates temporary work tables used to compare customer's current and future balances with the sum of the customer's open documents. This query never returns any results directly, however it MUST be rerun each time SP_ARDOCBAL_CUSTBAL is run.

   Resolution: Not applicable since the integrity check will not return errors.

2. **SP_ARDOC_STS** —Verify consistency of ARDOC statuses and balances

   Explanation: Identifies all accounts receivable documents where the document balance is inconsistent with the OPENDOC status. For example, when the document balance is zero, the document should be closed. When the document balance is non-zero, the document status should be open.

   Resolution: *Integrity Wizard* includes an automatic repair that updates all open documents that have a zero balance to a closed status. Documents that are closed but have a non-zero balance will have to be researched and manually repaired.

3. **SP_ARDOC_VOID_BATCH** — Identify non-voided transactions with a voided document

   Explanation: Returns all records in ARDOC where the document has been voided but the corresponding transaction has not been voided.

Resolution: Identified records should be researched further. Once researched and verified that the APTRANs are invalid, the automatic repair option in *Integrity Wizard* can be run to update the TRANTYPE in APTRAN to equal the corresponding DOCTYPE of VT.

4. **SP_ARDOCBAL_ARADJ** — Verify DOCBAL equals calculated DOCBAL in ARDOC

   Explanation: Fatal Error — Checks open ARDOCs to determine if DOCBAL is equal to the calculated document balance based on the associated records in APADJUST.

   Resolution: Any records returned must be researched and manually adjusted.

5. **SP_ARDOCBAL_CUSTBAL** — Check CUSTOMER current and future balances to ARDOC balances

   Explanation: Returns all instances where a customer's current and/or future balance does not equal the sum of the open document balances for the customer.

   Resolution: *Integrity Wizard* includes an automatic repair option that will update the customer's current balance and future balance to the calculated value.

6. **SP_ARTRAN_VOID_ARDOC** — Identify non-voided transactions with a voided document

   Explanation: Returns all records in ARDOC where the document has been voided but the corresponding transaction has not been voided.

   Resolution: Identified records should be researched further. Once researched and verified that the APTRANs are invalid, the automatic repair options in *Integrity Wizard* can be run to update the TRANTYPE in APTRAN to equal the corresponding DOCTYPE of VT.

7. **SP_ARTRAN_VOID_BATCH** — Identify non-voided transactions with a voided batch

   Explanation: Returns all records in ARTRAN where the transaction has not been voided but the corresponding BATCH record has a status of V or Voided.

   Resolution: Identified records should be researched further. Once researched and verified that the APTRANs are invalid, the automatic repair option in *Integrity Wizard* can be run to update TRANTYPE to VT.

8. **SP_CUST_NO_SOADDR** — Returns all customers with no default address

   Explanation: Fatal Error — Returns all customers with no default address.

   Resolution: *Integrity Wizard* includes an automatic repair to create SOADDRESS records for each customer ID returned using the address information contained in the CUSTOMER table.

9. **SP_DEL_INACT_CUST** — Check for inactive customers with no documents

Explanation: Identifies all customers with a status of One Time or Inactive that have no documents in the database. The result set is not an error but provides an opportunity to purge inactive customers through *Integrity Wizard*.

Resolution: *Integrity Wizard* will delete all One Time and Inactive customers that do not have any non-voided ARDOCs in the database.

10. **SP_INVAL_TERMS_ARDOC** — Identify ARDOCs with invalid TERMS codes

    Explanation: Fatal Error — Returns ARDOC records that contain blank or invalid TERMS values. Accounts receivable documents without a valid terms ID will not display on the *Aged AR* (08.610.00) report causing the appearance of missing documents.

    Resolution: Update each identified record with a valid terms ID.

11. **SP_INVAL_TERMS_CUSTOMER** — Identify customers with invalid terms codes

    Explanation: Fatal Error — Returns all customers that are using a terms code that is not contained in the TERMS table.

    Resolution: Identified records should be researched further. Once researched, the terms code should be entered in *Terms Maintenance* (21.270.00) or changed in *Customer Maintenance* (08.260.00).

12. **SP_NEG_DOCBAL_ARDOC** — Check for documents with a negative balance

    Explanation: Fatal Error — Returns all accounts receivable documents (open or closed) that have a document balance below zero.

    Resolution: Identified records should be researched to determine the correct document balance and status. Once determined, the DOCBAL field in ARDOC should be updated. The following SQL statement can be used:

    ```
    "UPDATE ARDOC SET DOCBAL = XXXX.XX
    WHERE CUSTID = 'XXXXXX' AND DOCTYPE =
    'XX' AND REFNBR = 'XXXXXX' AND BATNBR =
    'XXXXXX' AND BATSEQ = 'X';"
    ```

13. **SP_NLCURY_ARDOC** — Check for CURYIDs in ARDOC

    Explanation: Returns all ARDOCs where the DOCTYPE is not VT and the CURYID is null. This condition should not exist and probably indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the ARDOCs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the ARDOC is valid, the record must be updated to include the appropriate CURYID.

14. **SP_NLCURY_ARTRAN** — Check for CURYIDs in ARTRAN

    Explanation: Returns all ARTRANs with null CURYIDs. This condition should not exist and usually indicates a record that should be removed from the database.

Resolution: Identified records should be researched further. Once researched and verified that the ARTRANs are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that the ARTRAN is valid, the record must be updated to include the appropriate CURYID.

15. **SP_NLCUST_ARADJ** — Check for CUSTIDs in ARADJUST

    Explanation: Returns all ARADJUST records with null CUSTIDs. This condition should not exist and usually indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the ARADJUST records are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that an ARADJUST record is valid, the record must be updated to include the appropriate CUSTID.

16. **SP_NLCUST_ARDOC** — Check for CUSTIDs in ARDOC

    Explanation: Returns all ARDOC records with null CUSTIDs. This condition should not exist and usually indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the ARDOC records are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that an ARDOC record is valid, the record must be updated to include the appropriate CUSTID.

17. **SP_NLCUST_ARHIST** — Check for CUSTIDs in ARHIST

    Explanation: Returns all ARHIST records with null CUSTIDs. This condition should not exist and usually indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the ARHIST records are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that an ARHIST record is valid, the record must be updated to include the appropriate CUSTID.

18. **SP_NLCUST_ARTRAN** — Check for CUSTIDs in ARTRAN

    Explanation: Returns all ARTRAN records with null CUSTIDs. This condition should not exist and usually indicates a record that should be removed from the database.

    Resolution: Identified records should be researched further. Once researched and verified that the ARTRAN records are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that an ARTRAN record is valid, the record must be updated to include the appropriate CUSTID.

19. **SP_NLCUST_CUST** — Check for CUSTIDs in CUSTOMER

    Explanation: Returns all CUSTOMER records with null CUSTIDs. This condition should not exist and usually indicates a record that should be removed from the database.

Resolution: Identified records should be researched further. Once researched and verified that the CUSTOMER records are invalid, the automatic repair option in *Integrity Wizard* can be run to delete them. In the event that a CUSTOMER record is valid, the record must be updated to include the appropriate CUSTID.

20. **SP_NLYR_SLSPERHIST** — Check for blank FISCYR in SLSPERHIST

    Explanation: Return any SLSPERHIST records that have a blank value for the FISCYR field. Since this is an invalid record, the other values contained within it will display on screen or in reports.

    Resolution: Research with client to determine the correct SLSPERHIST record the blank one should be merged with and adjust records accordingly either via SQL or in Initialize mode in *Salesperson History* (08.311.00). The SQL query statements below may be used to assess the records. Once the invalid record amount has been correctly merged into a valid record year by updating the valid record year, the record with the blank FISCYR should be deleted using the following SQL statement:

    ```
    Delete from SlsPerHist where SlsPerID =
    "XXXXXX" and FiscYr = ""
    ```

To assess records:

```
Select SlsPerID, Fiscyr, YTDRcpt[-
ZZZ,ZZZ,ZZZ,ZZZ.99], YTDSls[-
ZZZ,ZZZ,ZZZ,ZZZ.99], YTDCogs[-
ZZZ,ZZZ,ZZZ,ZZZ.99], from SlsPerHist
where SlsPerID = "XXXXXX"
```

```
Select DocType, sum(OrigDocAmt)[-
ZZZ,ZZZ,ZZZ,ZZZ.99] from ARDoc where
SlsPerID = "XXXXXX" and
substring(perpost, 1, 4) = "YYYY" and
Rlsed = 'True' group by DocType
```

```
Select DocType, sum(OrigDocAmt)[-
ZZZ,ZZZ,ZZZ,ZZZ.99] from ARDoc where
SlsPerID = "BW" and substring(perpost,
1, 4) = "1996" and rlsed = 'True' group
by DocType;
```

```
Select TranType, sum(ExtCost)[-
ZZZ,ZZZ,ZZZ,ZZZ.99] from artran where
refnbr in (select refnbr from ARDoc
where SlsPerID = "XXXXXX" and
substring(perpost, 1, 4) = "YYYY" and
rlsed = 'True' ) and rlsed = 'True'
group by TranType
```

To determine which year might be off:

```
SlsPerHist.YTDRcpt = ARDoc.OrigDocAmt
where DocType = 'PA'
```

```
SlsPerHist.YTDSls = net of
ARDoc.OrigDocAmt where DocType = 'IN'
plus where DocType = 'DM' less where
Doctype = 'CM'
```

```
SlsPerHist.YTDCOGS = ARTRan.ExtCost
```

21.**SP_NOBATCH_ARDOC** — Check for orphan ARDOCs

Explanation: Identifies all ARDOCs without corresponding batch records. This condition should not exist in the database. The batch record not existing will prohibit the document from being retrieved in hand prepared *Invoice and Memo* (08.010.00) and on some reports. However, the documents will still display in Accounts Receivable's *Document Maintenance* (08.250.00) where batch numbers are not keys to displaying the document.

Resolution: *Integrity Wizard* offers two options to automatically resolve this database condition. The first option is to delete the identified ARDOC records. (Note that this option will delete even Open ARDOCS.) The second option provides the ability to create BATCH records for the orphaned ARDOCs. The options are presented after clicking **Next** in *Integrity Wizard* (IW.320.00).

22.**SP_NODOC_ARADJUSTD** — Check for accounts receivable adjusted DOCs without ARDOCs

Explanation: Identifies all ARADJUST records with a ADJDREFNBR that does not exist in ARDOC. Since the original REFNBR does not exist in ARDOC, it could potentially be created again with a duplicate number in the migrated database, causing a query and reporting issue.

Resolution: The risk of a duplicate being created is remote. This condition should be noted, however it is not as critical as the similar issue in Accounts Payable.

23. **SP_NODOC_ARADJUSTG** — Check for accounts receivable adjusting DOCs without ARDOCs

    Explanation: This query returns all ARADJUST records with an ADJGREFNBR that does not exist in ARDOC. Since the adjusting document is not in ARDOC, it may be created again with a duplicate number in the migrated database, causing a query and reporting issue.

    Resolution: The risk of a duplicate being created is remote. This condition should be noted, however it is not as critical as the similar issue in Accounts Payable.

24. **SP_NODOC_ARBATCH** — Check for accounts receivable batches with no DOCs

    Explanation: Identifies accounts receivable batches that have no documents. This condition may be the result of purging data.

    Resolution: Review identified batches and verify that they are invalid. *Integrity Wizard* includes an automatic repair that will set the BATCH status to Void for all identified batch records. The only harm this condition causes is an appearance of a data integrity issue when a user is running the Solomon IV v6.0 Accounts Receivable integrity check. In addition, if a user selects the batch in the respective edit screen, no documents will be returned.

25. **SP_NODOC_ARTRAN** — Check for orphan ARTRANs

    Explanation: Identifies ARTRAN records with no associated ARDOC. These records have no use in the database and should be deleted.

    Resolution: *Integrity Wizard* will delete all orphan ARTRANs with the exception of records with a TRANTYPE of DT and a DRCR of R. Records with these values do not have associated ARDOCs.

26. **SP_PERPOST_ARDOC_BATCH** — Compare PERPOST in ARDOC and BATCH

    Explanation: Fatal Error — Identifies ARDOC records where the PERPOST is not equal to the associated BATCH PERPOST. The period affected by posting is determined by the batch period-to-post and not by the document. However this problem will cause the *Period Sensitive Aged AR* (08.611.00) report to be misstated as well as other reporting issues.

    Resolution: *Integrity Wizard* includes an automatic repair that will update the ARDOC.PERPOST to equal the associated BATCH PERPOST.

27.**SP_PERPOST_ARTRAN_BATCH** — Compare PerPost in ARTRAN and BATCH

Explanation: Fatal Error — Identifies ARTRAN records where the PERPOST is not equal to the associated BATCH.PERPOST. The period affected by posting is determined by the batch period-to-post and not by the document or transactions. Unless repaired, all accounts receivable transactions with this condition will create general ledger transactions with the same problem. This issue will cause irregular results in the *Detail General Ledger* (01.620.00) report.

Resolution: *Integrity Wizard* includes an automatic repair that will update the ARTRAN PERPOST to equal the associated BATCH.PERPOST.

28.**SP_SOADDRESS_UQ** — Check for duplicate Customer Shipping Addresses

Explanation: Fatal Error — Identifies records in the SOADDRESS table with the same CUSTID/SHIPTOID combination. This situation is not permitted in Solomon IV version 6.0.

Resolution: *Integrity Wizard* includes an automatic repair that will delete the duplicate Customer Shipping Address records.

Cash Manager Integrity Checks

1. **SP_MAN_CARECUR** — Check for manual entries in CARECUR

   Explanation: Returns all manual entries in CARECUR where no recurrences have occurred.

   Resolution: *Integrity Wizard* includes an automatic repair to delete all entries in CARECUR where FRQOFGEN = Manual and NBRCYCLE = 0.

2. **SP_NULL_BANKREC** — Check for bank and currency data from BANKREC

   Explanation: Identifies records in BANKREC that have null values in either the BANKACCT or BANKSUB. All records in BANKREC should have valid values in BANKACCT and BANKSUB.

   Resolution: *Integrity Wizard* includes an automatic repair option that will delete identified records. It is recommended that the records be manually reviewed prior to deleting to ensure that a valid BANKACCT or BANKSUB cannot be inserted in these offending records.

3. **SP_NULL_CARECUR** — Check for bank and currency data from CARECUR

   Explanation: Identifies records in CARECUR that have null values in either the BANKACCT or BANKSUB. All records in CARECUR should have valid values in BANKACCT and BANKSUB.

   Resolution: *Integrity Wizard* includes an automatic repair option that will delete identified records. It is recommended that the records be manually reviewed prior to deleting to ensure that a valid BANKACCT or BANKSUB cannot be inserted in these offending records.

4. **SP_NULL_CASHACCT** — Check for bank and currency data from CASHACCT

   Explanation: Identifies records in CASHACCT that have null values in either the BANKACCT or BANKSUB. All records in CASHACCT should have valid values in BANKACCT and BANKSUB.

   Resolution: *Integrity Wizard* includes an automatic repair option that will delete identified records. It is recommended that the records be manually reviewed prior to deleting to ensure that a valid BANKACCT or BANKSUB cannot be inserted in these offending records.

5. **SP_NULL_CASHAVGD** — Check for bank and currency data from CASHAVGD auto repair

   Explanation: Identifies records in CASHAVGD that have null values in either the BANKACCT or BANKSUB. All records in CASHAVGD should have valid values in BANKACCT and BANKSUB.

   Resolution: *Integrity Wizard* includes an automatic repair option that will delete identified records. It is recommended that the records be manually reviewed prior to deleting to ensure that a valid BANKACCT or BANKSUB cannot be inserted in these offending records.

6. **SP_NULL_CASHFLOW** — Check for bank and currency data from CASHFLOW

   Explanation: Identifies records in CASHFLOW that have null values in either the BANKACCT or BANKSUB. All records in CASHFLOW should have valid values in BANKACCT and BANKSUB.

Resolution: *Integrity Wizard* includes an automatic repair option that will delete identified records. It is recommended that the records be manually reviewed prior to deleting to ensure that a valid BANKACCT or BANKSUB cannot be inserted in these offending records.

7. **SP_NULL_CASHSUMD** — Check for bank and currency data from CASHSUMD

   Explanation: Identifies records in CASHSUMD that have null values in either the BANKACCT or BANKSUB. All records in CASHASUMD should have valid values in BANKACCT and BANKSUB.

   Resolution: *Integrity Wizard* includes an automatic repair option that will delete identified records. It is recommended that the records be manually reviewed prior to deleting to ensure that a valid BANKACCT or BANKSUB cannot be inserted in these offending records.

8. **SP_NULL_CATRAN** — Check for bank and currency data from CATRAN

   Explanation: Identifies records in CATRAN that have null values in either the BANKACCT or BANKSUB. All records in CATRAN should have valid values in BANKACCT and BANKSUB.

   Resolution: *Integrity Wizard* includes an automatic repair option that will delete identified records. It is recommended that the records be manually reviewed prior to deleting to ensure that a valid BANKACCT or BANKSUB cannot be inserted in these offending records.

9. **SP_PERPOST_CATRAN_BATCH** — Compare PERPOST in CATRAN and BATCH

Explanation: Fatal Error — Identifies CATRAN records where the PERPOST is not equal to the associated BATCH PERPOST.

Resolution: *Integrity Wizard* includes an automatic repair that will update the CATRAN PERPOST to equal the associated BATCH.PERPOST.

Payroll Integrity Checks

1. **SP_PR_CHK_BAT_NODOC** — Check for orphan batches corresponding to PR checks

   Explanation: Checks for payroll batch records without corresponding checks in PRDOC. All payroll batches should have associated documents in PRDOC. Batches without corresponding PRDOC records should not be migrated.

   Resolution: *Integrity Wizard* includes an automatic repair that will delete identified BATCH records.

2. **SP_PR_CHK_STUB_NODOC** — Check for orphan STUBDETAIL

   Explanation: Checks for STUBDETAIL records without corresponding records in PRDOC. All STUBDETAIL records should have associated documents in PRDOC. STUBDETAIL records without corresponding PRDOC records should not be migrated.

   Resolution: *Integrity Wizard* includes an automatic repair that will delete orphaned STUBDETAIL records.

3. **SP_PR_CHK_TRAN_NODOC** — Check for orphan PRTRANs corresponding to PR checks

   Explanation: Checks for PRTRAN records without corresponding records in PRDOC. PRTRAN records with a TIMESHTFLG = 1 and a TYPE value of RC are excluded.

   Resolution: *Integrity Wizard* includes an automatic repair that will delete orphaned PRTRAN records where the TIMESHTFLG = 0 and TYPE <> RC.

4. **SP_PR_CHK_YEAREND** — Verify that Year-End Updates have been applied

Explanation: Checks the database to ensure that the latest year-end updates have been applied to the database. If these updates have not been applied, an error will occur in *Transformation Wizard* when payroll is being processed.

Resolution: *Integrity Wizard* includes an automatic repair that will apply the year-end updates against the source database.

5. **SP_PR_EARNDED_QTD_YTD** —Verify sum QTD = YTD for every EARNDED record

   Explanation: Verifies the sum of quarterly deductions equals the year-to-date value in the same record within the EARNDED table. Older versions of Solomon IV did not store four quarters of information which can result in QTD numbers that do not sum to YTD amounts.

   Resolution: Use *Employee History* (02.060.00) to initialize the correct QTD and YTD earnings and deductions for the employees identified. This is required if YTD earnings and deductions were initialized previously without also initializing the quarter-to-date amounts.

6. **SP_PR_EARNDED_DED** — Verify EARNDED refers to valid deductions

   Explanation: Identifies any EARNDEDID in the EARNDED table that does not have a corresponding DEDID in the DEDUCTION table. All EARNDEDID values should have corresponding values in the DEDUCTION master table.

   Resolution: Two resolutions are appropriate. Either update the EARNDED record with an EARNDEDID that has a corresponding value in the DEDUCTION table, or add a record to DEDUCTION to correspond with the value in EARNDEDID.

7. **SP_PR_EARNDED_EARN** — Verify EARNDED refers to valid earning types

   Explanation: Identifies any EARNDEDID in the EARNDED table that does not have a corresponding ID in the EARNTYPE table. All EARNDEDID values should have corresponding values in the EARNTYPE master table.

   Resolution: Two resolutions are appropriate. Either update the EARNDED record with an EARNDEDID that has a corresponding value in the EARNTYPE table, or add a record to EARNTYPE to correspond with the value in EARNDEDID.

8. **SP_PR_EXMPT_REDUCE** — Check for Solomon III migration DEDID-REDUCE inconsistencies

   Explanation: Check for inconsistencies where DEDID = P and reduce = TRUE from the EXMPTCREDIT table. This should only be a concern if migrating from Solomon III.

Resolution: *Integrity Wizard* includes an automatic repair to set REDRPTEARNSUBJDED = 0 where BASETYPE = P and REDRPTEARNSUBJDED = 1.

9. **SP_PR_NULL_PRDOC** — Check for NULLS in PRDOC

   Explanation: Identifies PRDOC records with null values in ACCT, SUB, CHKNBR, or DOCTYPE.

   Resolution: Identified records should be researched to determine if they are valid records. If the records are valid, update the null fields with valid values. If the records are not valid, they should be manually deleted.

10. **SP_PR_NULL_STUBDET** — Check for NULLS in STUBDETAIL

    Explanation: Identifies STUBDETAIL records with null values in ACCT, SUB, CHKNBR, DOCTYPE, TYPE, or TYPEID.

    Resolution: Identified records should be researched to determine if they are valid records. If the records are valid, update the null fields with valid values. If the records are not valid, they should be manually deleted.

11. **SP_PR_PAYPER_ORPH** — Check for orphan PAYPERIODs

    Explanation: Identifies PAYPERIOD records that do not have corresponding records in PRDOC and EMPLOYEE.

    Resolution: *Integrity Wizard* includes a repair option to automatically delete identified records from PAYPERIOD.

12. **SP_PR_REC_BAT_NOTRAN** — Check for orphan PR check reconciliation batches

    Explanation: Identifies check reconciliation batches that do not have a corresponding record in PRTRAN.

Resolution: *Integrity Wizard* includes a repair option to automatically delete identified records from BATCH.

13. **SP_PR_TSH_BAT_NOTRAN** — Check for orphan timesheet batches

    Explanation: Identifies timesheet batches that do not have a corresponding record in PRTRAN.

    Resolution: *Integrity Wizard* includes a repair option to automatically delete identified records from BATCH.

14. **SP_PR_ZEARN_CHK** — Verify EMPLOYEE.YTDEARN <>0 when at least one check exists for current calendar year

    Explanation: Verifies that at least one, non-zero check exists in PRDOC if an employee's year-to-date earnings are not equal to 0.00. Records are returned that have employees with 0.00 year-to-date earnings, with a PRDOC existing in the database for the same calendar year.

    Resolution: Use *Employee History* (02.060.00) to initialize the correct QTD and YTD earnings and deductions for the employees identified. This is required if YTD earnings and deductions were initialized previously without also initializing the QTD amounts. Older versions of Solomon IV did not store four quarters of information that can result in QTD numbers that do not sum to the YTD amounts. If a check run failed to release properly, the system may not have been properly updated.

Project Controller Integrity Checks

All database repairs for errors identified through Project Controller integrity checks must be resolved by analyzing the source database records and updating records through SQL Scope, Query Analyzer, or by using Initialization mode. *Integrity Wizard* does not include systematic repair for errors returned on Project Controller integrity checks.

1. **SP_PC_SUM** — Create Summaries for PC — Project Controller

   Explanation: Populates temporary work tables used for analyzing Project Series records. The check never returns any results directly, however it MUST be rerun each time any modifications are made to the source database as a result of errors identified through any of the Project Controller integrity checks. This integrity check is automatically selected to run if Process Flow is rebuilt (*Integrity Wizard* (IW.110.00)), alternatively, SP_PC_SUM can be selected by double-clicking **Analyze Project Controller** in *Integrity Wizard* (IW.120.00), and then selecting this specific integrity check from the list displayed.

   Resolution: Not applicable since errors are not returned with this check.

2. **SP_PJACCOUNT_ACCT** — Check for ACCT and GL_ACCT in PJ_ACCOUNT

   Explanation: Fatal Error — PJACCOUNT is the cross-reference between general ledger accounts and Project Series account categories. This returns records with blank ACCTs or GL_ACCTs in PJ_ACCOUNT.

Resolution: In most cases if there is a record with a blank ACCT or GL_ACCT, a valid replacement record already exists. Therefore, the first step in the resolution is to verify if a valid replacement record exists for the identified record. If a valid record does exist the record returned should be deleted. If a valid replacement record does not exist, then delete the record and update *Chart of Accounts Maintenance* (01.260.00). An entry in *Chart of Accounts Maintenance* (01.260.00) will update the PJ_ACCOUNT table.

3. **SP_PJCHARGD_DETNUM** — Check that DETAIL_NUM in PJCHARGD is small integer

   Explanation: Fatal Error — PJCHARGD stores the detail transaction records for a batch entered using *Project Charges* (PA.060.00. When posted, they are transferred to table PJTRAN. DETNUM is the line number used in PJCHARGD and is used as the unique key. This query verifies that all DETAIL_NUM values in PJCHARGD are small integers, therefore less than 32768.

   Resolution: Any identified records must be deleted from PJCHARGD as well as any other records in PJCHARGD associated with the batch. In addition, associated records in PJCHARGH must be deleted. The detail information will not be lost as it is stored in PJTRAN.

4. **SP_PJCHARGH_BATSTS** — Verify batch status in PJCHARGH

   Explanation: Fatal Error — Verifies that all records in PJCHARGH have a BATCH_STATUS = P. PJCHARGH stores the header records for all batches entered using *Project Charges* (PA.060.00).

   Resolution: Records not having a status of P will be returned. Identified batches must be posted or deleted.

5. **SP_PJEXPHDR_STS** — Verify status in PJEXPHDR

   Explanation: Fatal Error — Verifies that all records in PJEXPHDR have a STATUS_1 value = P. PJEXPHDR contains the travel and expense report header records. FastForward 6.0 requires that all travel and expense records have a status of Posted, prior to migration. This query applies to Advanced Edition only.

   Resolution: Records not having a status of P will be returned. Identified records must be posted or deleted.

6. **SP_PJINVHDR_INVSTS** —Verify invoice status in PJINVHDR

   Explanation: Fatal Error — Verifies that all records in PJINVHDR have an INV_STATUS value = PO. PJINVHDR contains the invoice header records. FastForward 6.0 requires that all invoices have a status of Posted, prior to migration. This query applies to Advanced Edition only.

   Resolution: Records not having a status of PO will be returned. Identified records must be posted or deleted.

7. **SP_PJLABDIS_STS** — Verify status in PJLABDIS

Explanation: Fatal Error — PJLABDIS contains the detail labor distribution records of a time card, after they are approved and posted. This query verifies that the detail labor distribution records of a time card, all have a status of Posted. This check is dependent on the setup information for Time and Expense for Projects. The query will only be performed if **Require GL Labor Posting to Close** is checked.

Note: This information is stored in the PJCONTRL table.

Resolution: Records not having a status of P will be returned. Identified records must be posted or deleted.

8. **SP_PJLABHDR_LESTS** — Verify LE_STATUS in PJLABHDR

   Explanation: Fatal Error — PJLABHDR contains the Time Card Header records. This query verifies that the LE_STATUS of all PJLABHDR records = Posted, or X, timecards that have been corrected.

   Resolution: Records not having a status of P or X will be returned. Identified records must be posted or deleted.

9. **SP_PJPAYHDR_STS** — Verify status in PJPAYHDR

   Explanation: Fatal Error — PJPAYHDR contains Subcontractor Payment Header records. Each record in this table represents a specific payment request of a subcontract to a vendor. As well as containing header information identifying the payment, records in this table contain references to the accounts payable vouchers that were created to pay this request. This query verifies that all records in PJPAYHDR have a STATUS1 value = Posted. This query applies to Advanced Edition only.

   Resolution: Records not having a status of P will be returned. Identified records must be posted or deleted.

10. **SP_PJPENT_MISSING** — Compare PJPENT and PJPENTEX tables for corresponding records

    Explanation: Fatal Error — PJPENT and PJPENTEX must contain corresponding records. This query will check for instances where there are PJPENTEX records that do not have corresponding PJPENT records.

    Resolution: An automatic repair is available in *Integrity Wizard*. The repair will delete orphaned PJPENTEX records that do not have corresponding PJPENT record.

11. **SP_PJPENTEX_MISSING** — Compare PJPENT and PJTRANEX tables for corresponding records

    Explanation: Fatal Error — PJPENT and PJPENTEX must contain corresponding records. This query will check for instances where there are PJPENT records that do not have corresponding PJPENTEX records.

    Resolution: An automatic repair is available in *Integrity Wizard*. The repair will insert new PJPENTEX records so that PJPENT and PJPENTEX continue to have a one-to-one relationship.

12. **SP_PJPROJ_MISSING** — Compare PJPROJ and PJPROJEX tables for corresponding records

    Explanation: Fatal Error — PJPROJ and PJPROJEX must contain corresponding records. This query will check for instances where there are PJPROJEX records that do not have corresponding PJPROJ records.

    Resolution: An automatic repair is available in *Integrity Wizard*. The repair will delete orphaned PJPROJEX records that do not have corresponding PJPROJ records.

13. **SP_PJPROJEX_MISSING** — Compare PJPROJ and PJTRANEX tables for corresponding records

    Explanation: Fatal Error — PJPROJ and PJPROJEX must contain corresponding records. This query will check for instances where there are PJPROJ records that do not have corresponding PJPROJEX records.

    Resolution: An automatic repair is available in *Integrity Wizard*. The repair will insert new PJPROJEX records so that PJPROJ and PJPROJEX continue to have a one-to-one relationship.

14. **SP_PJREVHDR_STS** — Verify status in PJREVHDR

    Explanation: Fatal Error — PJREVHDR contains the header record of a budget revision. At this level, the revision ID, project, revision type, approvers, and change order are defined. This query verifies that all PJREVHDR records have STATUS = P, posted. This query applies to Advanced Edition only.

    Resolution: Records not having a status of P will be returned. Identified records must be posted or deleted.

15. **SP_PJTRAN_MISSING** — Compare PJTRAN and PJTRANEX tables for corresponding records

Explanation: Fatal Error — PJTRAN and PJTRANEX must contain corresponding records. This query will check for instances where there are PJTRANEX records that do not have corresponding PJTRAN records.

Resolution: An automatic repair is available in *Integrity Wizard*. The repair will delete orphaned PJTRANEX records that do not have corresponding PJTRAN.

16. **SP_PJTRANEX_MISSING** — Compare PJTRAN and PJTRANEX tables for corresponding records

    Explanation: Fatal Error — PJTRAN and PJTRANEX must contain corresponding records. This query will check for instances where there are PJTRAN records that do not have corresponding PJTRANEX records.

    Resolution: An automatic repair is available in *Integrity Wizard*. The repair will insert new PJTRANEX records so that PJTRAN and PJTRANEX continue to have a one-to-one relationship.

17. **SP_PJTRAN_PJACTROL** — Check that PJTRAN rolls up to summary in PJACTROL

    Explanation: PJACTROL is a summary of actual balances by project-account category. It has one record per fiscal year that stores each fiscal period's amount and quantity in separate fields. This query utilizes temporary tables created by SP_PC_SUM to verify that the sum of PJTRAN AMOUNTs and UNITs equal the values in PJACTROL AMOUNTs and UNITs for each project-account category. Projects are returned in cases where the sum of AMOUNTs or UNITs does not equal the corresponding values in PJACTROL.

Resolution: Project Series has a utility available on Microsoft Business Solutions' FTP server to repair the summary tables. Download the file BSUTBLD.ZIP (located at ftp://ftp.solomon.com/s4/PMG/PMG20x/). It is important to read and follow the instructions with this utility carefully. Failure to do so could potentially cause unexpected results.

18. **SP_PJTRAN_PJACTSUM** — Check that PJTRAN rolls up to summary in PJACTSUM

    Explanation: PJACTSUM is a summary of actual balances by project-task-account category. It has one record per fiscal year that stores each fiscal period's amount and quantity in separate fields. This query utilizes temporary tables created by SP_PC_SUM to verify that the sum of PJTRAN AMOUNTs and UNITs equal the values in PJACTSUM AMOUNTs and UNITs for each project-task-account category. Projects and tasks are returned in cases where the sum of AMOUNTs or UNITs does not equal the corresponding values in PJACTSUM.

    Resolution: Project Series has a utility available on Microsoft Business Solutions' FTP server to repair the summary tables. Download the file BSUTBLD.ZIP (located at ftp://ftp.solomon.com/s4/PMG/PMG20x/). It is important to read and follow the instructions with this utility carefully. Failure to do so could potentially cause unexpected results.

19. **SP_PJTRAN_PJPTDROL** — Check that PJTRAN rolls up to summary in PJPTDROL

    Explanation: PJPTDROL is the primary summary file for project balances. It has one record per project + account category, and contains the current balance for actuals, revised budgets (EAC), original budgets and commitments.

    This query utilizes temporary tables created by SP_PC_SUM to verify that the sum of PJTRAN AMOUNTs and UNITs equal the values in PJPTDROL AMOUNTs and UNITs for each project-account category. Projects are returned in cases where the sum of AMOUNTs or UNITs does not equal the corresponding values in PJPTDROL.

Resolution: Project Series has a utility available on Microsoft Business Solutions' FTP server to repair the summary tables. Download the file BSUTBLD.ZIP (located at ftp://ftp.solomon.com/s4/PMG/PMG20x/). It is important to read and follow the instructions with this utility carefully. Failure to do so could potentially cause unexpected results.

20. **SP_PJTRAN_PJPTDSUM** — Check that PJTRAN rolls up to summary in PJPTDSUM

    Explanation: PJPTDSUM is the primary summary file for task balances. It has one record per project + task + account category, and contains the current balance for actuals, revised budgets (EAC), original budgets and commitments. This query utilizes temporary tables created by SP_PC_SUM to verify that the sum of PJTRAN AMOUNTs and UNITs equal the values in PJPTDSUM AMOUNTs and UNITs for each project-task-account category. Projects are returned in cases where the sum of AMOUNTs or UNITs does not equal the corresponding values in PJPTDSUM.

    Resolution: Project Series has a utility available on Microsoft Business Solutions' FTP server to repair the summary tables. Download the file BSUTBLD.ZIP (located at ftp://ftp.solomon.com/s4/PMG/PMG20x/). It is important to read and follow the instructions with this utility carefully. Failure to do so could potentially cause unexpected results.

21. **SP_PJTRANEX_UQ** — Check for duplicate TR_ID11 in PJTRANEX

    Explanation: Fatal Error — Returns records with duplicate TR_ID11 values. There is an index on this field in Solomon IV v6.0 and therefore any duplicate values must be resolved.

Resolution: This field is created by concatenating the FISCALNO, SYSTEM_CD, BATCH_ID, and DETAIL_NUM fields from that same record. Always add leading zeroes to the value found in DETAIL_NUM when concatenating into TR_ID11. There are 10 positions reserved for DETAIL_NUM in TR_ID11, so leading zeroes should be added appropriately, including a minus sign if needed. For example, if the DETAIL_NUM value is -112, then TR_ID11 field would be -000000112.

22.**SP_PJEXPHDR_STS** — Verify Status in PJEXPHDR

Explanation: Fatal Error — Verifies that all records in PJEXPHDR have a STATUS_1 value of P. PJEXPHDR contains the travel and expense report header records. FastForward 6.0 requires that all travel and expense records have a status of P (posted) prior to migration. This query applies to Advanced Edition only.

Resolution: Records not having a status of P will be returned. These records must be posted or deleted.

23. **SP_PJINVHDR_INVSTS** — Verify Invoice Status in PJINVHDR

    Explanation: Fatal Error — Verifies that all records in PJINVHDR have a INV_STATUS value of PO. PJINVHDR contains the invoice header records. FastForward 6.0 requires that all invoices have a status of PO (posted) prior to migration. This query applies to Advanced Edition only.

    Resolution: Records not having a status of PO will be returned. These records must be posted or deleted.

24. **SP_PJINVTXT_DRAFTNUM** — Check for characters in DRAFT_NUM column in PJINVTXT table

    Explanation: Returns PJINVTXT records that contain non-numeric values in the DRAFT_NUM field.

    Resolution: Identified records should be researched further. Once researched, DRAFT_NUM should be populated with only a numeric value or the record should be deleted from the database.

25. **SP_PJPAYHDR_STS** — Verify Status in PJPAYHDR

    Explanation: Fatal Error — PJPAYHDR contains Subcontractor Payment Header records. Each record in this table represents a specific payment request of a subcontractor to a vendor. As well as containing header information identifying the payment, records in this table contain references to the accounts payable vouchers that were created to pay this request. This query verifies that all records in PJPAYHDR have a STATUS1 value of P (posted). This query applies to Advanced Edition only.

    Resolution: Records not having a status of P will be returned. These records must be posted or deleted.

26. **SP_PJREVHDR_STS** — Verify Status in PJREVHDR

Explanation: Fatal Error — PJREVHDR contains the header record of a budget revision. At this level, the revision ID, project, revision type, approvers, and change order are defined. This query verifies that all PJREVHDR records have a STATUS of P (posted). This query applies to Advanced Edition only.

Resolution: Records not having a status of P will be returned. These records must be posted or deleted.

Inventory and Purchasing Integrity Checks

1. **SP_ADJ_NEGTRANAMT** — Check for situations that will create a sign inconsistency in INTRAN

   Explanation: Fatal Error — Reviews the ITEMCOST and ITEMSITE tables for sign inconsistencies (negative quantity and positive cost or vice versa). This situation would cause an INTRAN record to be created in the Solomon IV version 6.0 database in the same manner which causes problems if a rebuild is subsequently run in Solomon IV version 6.0.

   Resolution: Identified records must be manually repaired in the source database so that sign inconsistencies no longer exist.

2. **SP_IN_SUM** — Create summaries for IN

   Explanation: Populates temporary work tables used for analyzing inventory records. The check never returns any results directly, however it MUST be rerun each time any modifications are made to the source database as a result of errors identified through any of the inventory integrity checks. This integrity check is automatically selected to run if Process Flow is rebuilt (*Integrity Wizard* (IW.110.00)), alternatively, SP_IN_SUM can be selected by double-clicking **Analyze Inventory** in *Integrity Wizard* (IW.120.00), and then selecting this specific integrity check from the list displayed.

   Resolution: Not applicable since errors are not returned.

3. **SP_INVAL_NEGQTY** — Check for invalid negative QTYONHAND in ITEMSITE

   Explanation: Checks for inventory items using the FIFO or LIFO valuation method with a negative QTYONHAND value in ITEMSITE.

Resolution: Identified records can be manually corrected in the source database or automatically corrected by running the *IN Integrity Check* (10.990.00) in the Solomon IV version 6.0 database using the Rebuild Item Quantities and Costs option.

4. **SP_LOTSER_NON_SERASSIGN** — Check for lot/serial tracked items with no SERASSIGN

   Explanation: Fatal Error — Check for inventory items that are marked as lot/serial tracked and have no assignment method. This condition should not exist and probably indicates that a record should be removed from the database.

   Resolution: Identified items should be researched further to determine a serial assignment method. Once determined, the assignment should be made using *Inventory Maintenance Defaults* (10.250.01). If it is determined that the item is no longer used, the record may be deleted from the INVENTORY table in the database.

5. **SP_NLKITID_KIT** — Check for null KITIDs in KIT

   Explanation: Returns all records in the KIT table where the KITID is null. This condition should not exist and usually indicates that a record should be removed from the database.

   Resolution: *Integrity Wizard* includes an automatic repair option to delete these records from the KIT table.

6. **SP_ NLKITID _COMPONENT** — Check for null KITIDs in COMPONENT

   Explanation: Returns all records in the COMPONENT table where the KITID is null. This condition should not exist and usually indicates that a record should be removed from the database.

   Resolution: *Integrity Wizard* includes an automatic repair option to delete these records from the COMPONENT table.

7. **SP_NLKITSITEID_COMPONENT** — Check for null KITSITEIDs in COMPONENT

   Explanation: Returns all records in the COMPONENT table where the KITSITEID is null. This condition should not exist and usually indicates that a record should be removed from the database.

   Resolution: *Integrity Wizard* includes an automatic repair option to delete these records from the COMPONENT table.

8. **SP_NLKITSTATUS_ COMPONENT** — Check for null KITSTATUS in COMPONENT

   Explanation: Returns all records in the COMPONENT table where the KITSTATUS is null. This condition should not exist and usually indicates that a record should be removed from the database.

Resolution: *Integrity Wizard* includes an automatic repair option to delete these records from the COMPONENT table.

9. **SP_NLSITEID_ITEMSITE** — Check for null SITEIDs in ITEMSITE

   Explanation: Fatal Error — Returns all records in ITEMSITE where the SITEID is null. This condition should not exist and usually indicates that a record should be removed from the database.

   Resolution: Identified records should be researched further. Once researched and verified, either the correct SITEID should be updated in the ITEMSITE table or the record should be removed.

10. **SP_NON_LOTSER_SERASSIGN** — Check for non-lot/serial tracked items with SERASSIGN

    Explanation: Fatal Error — Returns all inventory items that are not marked as lot/serial tracked (LOTSERTRACK = N) and have an assignment method.

    Resolution: *Integrity Wizard* includes an automatic repair to set the SERASSIGN field in INVENTORY to N (no assignment) for items not marked as lot/serial tracked that have an assignment method.

11. **SP_NONSTOCK_LOTSER** — Check for non-stock items set as lot/serial tracked

    Explanation: Returns all non-stock items set as lot/serial tracked.

    Resolution: *Integrity Wizard* includes an automatic repair to set non-stock items as non-serialized or non-tracked.

12. **SP_QTY_LOC_CST_LYR** — Verify quantity of receipt cost layered items for each location grouped by site

Explanation: Fatal Error — Verifies by site that the quantity for each item in the ITEMCOST table and LOCATION table are equal for cost layered F, FIFO, and L, LIFO, inventory items. In order for the data to migrate correctly, the quantity values in the ITEMSITE and ITEMCOST tables must be equal for each unique INVTID, SITEID combination.

Resolution: Research with client to determine if the ITEMCOST table or LOCATION table is correct and adjust accordingly using SQL Scope. The *Inventory Valuation* (10.620.00) and *Inventory Trial Balance* (10.630.00) reports can also be valuable tools in determining which table needs to be updated. Direct SQL query statements to obtain the existing values in each table follow. These queries should be compared by grouping the InvtID and SiteID combinations for each table to verify the quantity matches.

```
Select InvtID, SiteID, RcptNbr,
RcptDate, Qty from ItemCost where
InvtID = "XXXXXX"
```

```
Select InvtID, SiteID, LotSerNbr,
WhseLoc, QtyonHand from Location where
InvtID = "XXXXXX"
```

13. **SP_QTY_PER_ID** — Verify quantity of items per specific ID

Explanation: Fatal Error — Verifies by location that the quantity for each item in the LOCATION table and ITEMCOST table are equal for items with a valuation method of S, Specific ID. In order for the data to migrate correctly, the quantity values in the ITEMCOST and LOCATION tables must be equal for each unique INVTID, SITEID, and LOTSERNBR combination.

Resolution: Research with client to determine if the location or item cost values are correct and adjust accordingly using SQL Scope. The *Inventory Valuation* (10.620.00) and *Inventory Trial Balance* (10.630.00) reports are valuable tools in determining which table needs to be updated. The SQL query statements listed below can be used to obtain existing values in each of the tables. These queries should be compared by grouping the INVTID, SITEID, and LOTSERNBR combinations for each table to verify the quantity matches.

```
Select InvtID, SiteID, LotSerNbr, Qty
from ItemCost where InvtID = 'XXXXXX'
```

```
Select InvtID, SiteID, LotSerNbr,
QtyonHand from Location where InvtID =
'XXXXXX'
```

14. **SP_QTY_PER_SITE** — Verify quantity of items for each

    Explanation: Fatal Error — Verifies by SITE that the quantity for each item in the ITEMSITE table and LOCATION table are equal for all inventory items. In order for the data to migrate correctly, the quantity values in the ITEMSITE and LOCATION tables must be equal for each unique INVTID, SITEID combination.

    Resolution: Research with client to determine if the ITEMSITE table or LOCATION table is correct and adjust accordingly using SQL Scope. The *Inventory Valuation* (10.620.00) and *Inventory Trial Balance* (10.630.00) reports are valuable tools in determining which table needs to be updated. The SQL query statements listed below can be used to obtain existing values in each of the tables. These queries should be compared by grouping the INVTID and SITEID combinations for each table to verify the quantity matches.

    ```
    Select InvtID, SiteID, QtyOnHand from
    ItemSite where InvtID = 'XXXXX'
    ```

    ```
    Select InvtID, SiteID, LotSerNbr,
    WhseLoc, QtyOnHand from Location where
    InvtID = 'XXXXX'
    ```

15. **SP_QTY_SUM_CST_LYR** — Verify quantity for each cost layered item equals the site quantity

    Explanation: Fatal Error — Verifies by site that the quantity for each item in the ITEMSITE table and ITEMCOST table are equal for cost layered items with valuation methods of S, Specific ID, F, FIFO, or L LIFO. In order for the data to migrate correctly, the cost values in the ITEMSITE and ITEMCOST tables must be equal for each unique INVTID, SITEID combination.

Resolution: Research with client to determine if the ITEMCOST or ITEMSITE values are correct and adjust accordingly using SQL Scope. The *Inventory Valuation* (10.620.00) and *Inventory Trial Balance* (10.630.00) reports can also be valuable tools in determining which table needs to be updated. Direct SQL query statements to obtain the existing values in each table follow. These queries should be compared by grouping the InvtID, SiteID combinations for each table to verify the quantity matches.

```
Select InvtID, SiteID, LotSerNbr,
RcptNbr, RcptDate, Qty from ItemCost
where InvtID = "XXXXXX"
```

```
Select InvtID, SiteID, QtyonHand from
ItemSite where InvtID = "XXXXXX"
```

16. **SP_TOTCOST_PER_SITE** — Verify total cost of items for each site

    Explanation: Fatal Error — Verifies that TOTCOST for an item in ITEMSITE equals the TOTCOST for the item in ITEMCOST for all inventory items with a valuation method of FIFO, LIFO, or Specific ID. In order for the data to migrate correctly, the cost values in the ITEMSITE and ITEMCOST tables must be equal for each unique INVTID, SITEID combination.

    Resolution: Research with client to determine if the ITEMSITE table or ITEMCOST table is correct and adjust accordingly using SQL Scope. The *Inventory Valuation* (10.620.00) and *Inventory Trial Balance* (10.630.00) reports are valuable tools in determining which table needs to be updated. The SQL query statements listed below can be used to obtain existing values in each of the tables. These queries should be compared by grouping the INVTID and SITEID combinations for each table to verify the quantity matches.

    ```
    Select InvtID, SiteID, LotSerNbr,
    RcptNbr, RcptDate, TotCost from
    ItemCost where InvtID = 'XXXXX'
    ```

    ```
    Select InvtID, SiteID, TotCost from
    ItemSite where InvtID = 'XXXXX'
    ```

17. **SP_HOLD_PURCH_BATCH** — Check purchasing batches on hold

Explanation: Fatal Error — Identifies accounts payable batches on hold that were created by association with a purchase order through the automatic vouchering process in *Receipt/Invoice Entry* (04.010.00), or by entering a valid purchase order number in *Voucher and Adjustment Entry* (03.010.00). *Integrity Wizard* requires that all batches be released prior to migrating. In cases where vouchers were created from receipts, it may not be desirable to release the batches if the actual invoice for the items is not yet received from the vendor. Therefore, *Integrity Wizard* includes this check to identify these batches so that they can be handled separately.

Resolution: *Integrity Wizard* includes an automatic repair that will delete the identified batches from the database. However, prior to running the repair it is important to print the Excel spreadsheet created by *Integrity Wizard* for this integrity check. This spreadsheet should be retained so that the identified batches can be re-entered in the Solomon IV v6.0 database after migration. Batches can be re-entered in *Voucher and Adjustment Entry* (03.010.00) by selecting the purchase order number and receipt number from the purchase order number possible values list. There is a scenario that will prevent the purchase order number and receipt number from being selected. The defaults for Purchasing in *Integrity Wizard* require that a Vouchering Stage be assigned to all purchase orders. Purchase orders that are set to fully vouchered/fully received cannot be selected from the purchase order number possible values list and must be manually re-entered.

Order Management Integrity Checks

1. **SP_BOCNTR_IS_1** — Check for repairable BackOrders with a BOCNTR of 0

   Explanation: Fatal Error — Returns all records from SLSORDDET where BOCNTR is 0 but the corresponding SALESORD record's BOCNTR is 1.

   Resolution: *Integrity Wizard* includes an automatic repair to set the BOCNTR value in SLSORDDET to 1 for these records.

2. **SP_BOCNTR_OVER_1** — Check for non-repairable BackOrders with a BOCNTR of 0

   Explanation: Fatal Error — Returns all records from SLSORDDET where BOCNTR is 0 but the corresponding SALESORD record's BOCNTR is greater than 1.

   Resolution: Identified records should be researched further. Once researched and verified, the BOCNTR in SLSORDDET should be changed so that it corresponds to the BOCNTR value in SALESORD.

3. **SP_CREDITHOLD_OM** — Check for Customers with Credit Hold status

   Explanation: Identifies customers with a status of Credit Hold. Order Management no longer uses the Credit Hold Status.

   Resolution: *Integrity Wizard* includes an automatic repair option to set ALL customers with a status of Credit Hold to either Active or Inactive.

4. **SP_NLORDTP_SLSORD** — Check for blank order types in SALESORD

Explanation: Returns all records from SALESORD where ORDTYPE is blank or null. This condition should not exist and usually indicates a record that should be removed from the database.

Resolution: Identified records should be research further. Once researched and verified that the sales order record is invalid, the automatic repair option in *Integrity Wizard* can be run to delete all sales orders with a blank or null ORDTYPE.

5. **SP_NLORDTP_SLSORDDET** — Check for blank order types in SLSORDDET

   Explanation: Returns all records from SLSORDDET where ORDTYPE is blank or null. This condition should not exist and usually indicates a record that should be removed from the database.

   Resolution: Identified records should be researched further. Once researched and verified that the detail record is invalid, the automatic repair option in *Integrity Wizard* can be run to delete all sales order detail records with a blank or null ORDTYPE.

6. **SP_OP_SUM** — Create summaries for order processing

   Explanation: Populates temporary work tables used for analyzing order processing records. The check never returns any results directly, however it MUST be rerun each time any modifications are made to the source database as a result of errors identified through the SP_SLSORD_BAL integrity check. This integrity check is automatically selected to run if Process Flow is rebuilt (*Integrity Wizard* (IW.110.00)), alternatively, SP_OP_SUM can be selected by double-clicking **Analyze Order Processing** in *Integrity Wizard* (IW.120.00), and then selecting this specific integrity check from the list displayed.

   Resolution: Not applicable since the integrity check does not return errors.

7. **SP_SLSORD_BAL** — Verify order balances

   Explanation: Verifies both the calculated order total equals the specified order total and that the calculated invoice total equals the specified invoice total per sales order.

   Resolution: If the order is still an open order, then the order should be opened within *Sales Order and Memo* (05.260.00) and inspected for changes and resaved to correctly update the totals.

8. **SP_SLSORDDET_UQ** — Check for duplicates in SLSORDDET

   Explanation: Fatal Error — Checks for duplicate SLSORDDET records based on the combination of ORDNBR, BOCNTR, LINENBR, and ORDTYPE.

Resolution: Identified records should be researched further. Once researched, the duplicate records should be deleted or modified such that a duplicate will no longer exist.

9. **SP_ORD_ON_HOLD** — Check orders on hold

   Explanation: Fatal Error — Returns any orders with a status of Admin Hold (H) or Credit Hold (R). Orders in the source database cannot have a status of Admin Hold (H) or Credit Hold (R) prior to migration.

   Resolution: Orders should either be removed from a Hold status or deleted from within Solomon in *Sales Order and Memo* (05.260.00). After making corrections, run the *Order History* (05.670.00) report using the sort/select criteria of SalesOrd.Status; In; R, H to verify all orders have been cleared.

10. **SP_ORD_CA_OPEN** — Check open Cash Sale orders

    Explanation: Fatal Error — Returns any cash sale orders that are not closed. FastForward 6.0 does not migrate Cash Sale (CA) orders, therefore all orders with an order type of CA must be closed prior to the migration.

    Resolution: All cash sale orders should be either processed completely until Closed or deleted from within Solomon in *Sales Order and Memo* (05.260.00). After making corrections, run the *Order History* (05.670.00) report using the sort/select criteria below to verify all orders have been cleared.

    - 1st line — SalesOrd.OrdType; Equal; CA
    - 2nd line — SalesOrd.Status; Not Equal; C.

11. **SP_ORD_CM_OPEN** — Check open credit memo orders

Explanation: Fatal Error — Returns any credit memo orders that are not closed. FastForward 6.0 does not migrate Credit Memo (CM) orders, therefore all orders with an order type of CM must be closed prior to migration.

Resolution: All Credit Memo orders should be either processed completely until Closed or deleted from within Solomon in *Sales Order and Memo* (05.260.00). After making corrections, run the *Order History* (05.670.00) report using the sort/select criteria below to verify all orders have been cleared.

- 1st line — SalesOrd.OrdType; Equal; CM
- 2nd line — SalesOrd.Status; Not Equal; C

Service Series Integrity Checks

1. **SP_EMPPRID_UQ** — Check for unique EMPLOYEEPAYROLLID in XSMEMP

   Explanation: Returns all duplicate EMPLOYEEPAYROLLIDs in XSMEMP. This condition usually indicates that an employee has more than one EMPLOYEEID in the XSMEMP table.

   Resolution: Identified records should be researched further to determine the correct EMPLOYEEID and EMPLOYEEPAYROLLID for the employee. Once determined, the records with invalid EMPLOYEEPAYROLLIDs should either be removed or the EMPLOYEEPAYROLLID should be changed. If nothing is done to correct this, the most current EMPLOYEEID will be used to populate the SMSERVDETAIL table during *Transformation Wizard*.

2. **SP_INVAL_INVTID_FRDETAIL** — Check for invalid INVTIDs in XSMFRDETAIL

   Explanation: Returns INVTIDs in XSMFRDETAIL that are not listed in the INVENTORY table. This condition should not exist and probably indicates that a record was removed from the INVENTORY table after the returned record was created.

   Resolution: Identified records should be researched to determine the correct INVTID that should be associated with the record. This can also be resolved by adding the record’s INVTID to the INVENTORY table.

3. **SP_INVAL_INVTID_LBDETAIL** — Check for invalid INVTIDs in XSMLBDETAIL

Explanation: Returns INVTIDs in XSMLRDETAIL that are not listed in the INVENTORY table. This condition should not exist and probably indicates that a record was removed from the INVENTORY table after the returned record was created.

Resolution: Identified records should be researched to determine the correct INVTID that should be associated with the record. This can also be resolved by adding the record's INVTID to the INVENTORY table.

4. **SP_INVAL_INVTID_TMDETAIL** — Check for invalid INVTID in XSMTMDETAIL

   Explanation: Returns INVTIDs in XSMLRDETAIL that are not listed in the INVENTORY table. This condition should not exist and probably indicates that a record was removed from the INVENTORY table after the returned record was created.

   Resolution: Identified records should be researched to determine the correct INVTID that should be associated with the record. This can also be resolved by adding the record's INVTID to the INVENTORY table.

5. **SP_NL_MODELID** — Check for service equipment with no MODELID

   Explanation: Returns all service equipment items that do not have an associated model.

   Resolution: Identified records should be researched further to determine the correct MODELID to be associated with the service equipment item. Once the MODELID is determined, you must enter the model in *Manufacturer/Model Maintenance* (SM.106.00). The model then must be linked to the service equipment item in *Equipment Maintenance* (SM.107.00).

6. **SP_UQ_XSMEMP** — Check for duplicate employee IDs in XSMEMP

   Explanation: Fatal Error — Returns all records in XSMEMP that contain duplicate EMPLOYEEIDs. This condition should not exist and usually indicates a record that should be removed from the database.

   Resolution: Identified records should be researched further. Once researched and verified that the record is invalid, it should be removed from the XSMEMP table.

Bill of Material Integrity Checks

1. **SP_INVAL_KITTYPE** — Check for invalid TYPE in KIT

   Explanation: Fatal Error — Returns all records in the KIT table with an invalid TYPE value. Valid values are B or '' (blank).

   Resolution: Identified records should be researched further to determine if they are valid records. If the records are valid, update the TYPE with the correct value. If the records are not valid, they should be manually deleted.

2. **SP_NL_KEY_KIT** — Check for null keys in KIT

   Explanation: Fatal Error — Returns all kits that have a null value in the KITID, SITEID, or STATUS fields of the KIT table.

   Resolution: *Integrity Wizard* includes an automatic repair that removes any value from KIT that contains a null value in the KITID, SITEID, or STATUS fields.

3. **SP_NOKIT_COMPNT** — Check for components with no associated kit

   Explanation: Returns all components that are not associated with a kit.

   Resolution: This condition should not exist and usually indicates that a record should be deleted. *Integrity Wizard* includes an automatic repair that removes the returned components from the COMPONENT table.

4. **SP_NOKIT_ROUTING** — Check for routings with no kit

   Explanation: Returns all routings that are not associated with a kit.

Resolution: This condition should not exist and usually indicates that a record should be deleted. *Integrity Wizard* includes an automatic repair to remove the returned routings from the ROUTING table.

5. **SP_NOROUT_RTGSTEP** — Check for routing steps with no routings

   Explanation: Returns all routing steps with no associated routing.

   Resolution: This condition should not exist and usually indicates that a record should be deleted. *Integrity Wizard* includes an automatic repair to remove the returned routing steps from RTGSTEP.

Currency Manager Integrity Checks

1. **SP_ACCTHIST_NO_CURYACCT** — Check for foreign records with no translations

   Explanation: Fatal Error — Returns all foreign currency ACCTHIST records with no corresponding CURYACCT record.

   Resolution: Identified records should be researched further to determine if they are valid records. If the records are valid, a corresponding CURYACCT record must be inserted for each record. If the records are not valid, they should be manually deleted.

2. **SP_CURYACCT_GLTRAN_CURY** — Compare CURY GLTRAN and CURYACCT totals for prior years

   Explanation: Account/subaccount combinations are analyzed for prior fiscal years where the sum of general ledger transactions does not tie to the period to date balance in CURYACCT for foreign currencies.

   Resolution: Irregular conditions do not have to be repaired prior to the live migration. However, repair is recommended since not repairing them will cause each instance to be present on the general ledger integrity check in the migrated database. The repair must be done manually through SQL Scope or journal entries.

3. **SP_CURYACCT_GLTRAN_CURY_CRYR** — Compare CURY GLTRAN and CURYACCT totals for current year.

   Explanation: Fatal Error — Account/subaccount combinations are analyzed for the current fiscal year where the sum of general ledger transactions does not tie to the period to date balance in CURYACCT for foreign currencies.

Resolution: This condition must be repaired prior to the live migration. The repair must be done manually through SQL Scope or journal entries.

4. **SP_CURYACCT_NO_ACCTHIST** — Check for orphan translations

   Explanation: Fatal Error — Returns all CURYACCT records with no corresponding ACCTHIST record.

   Resolution: Identified records should be researched further to determine if they are valid records. If the records are valid, a corresponding ACCTHIST record must be inserted for each record. If the records are not valid, they should be manually deleted.

Using FastForward Database Creation

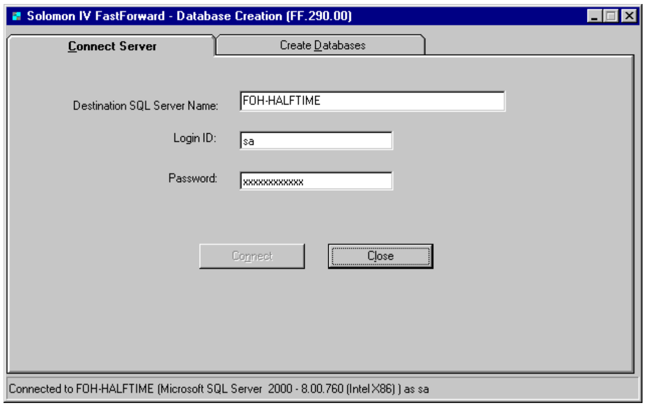
FastForward Database Creation Overview

Solomon IV FastForward - *Database Creation* (FF.290.00) MUST be used to create the Solomon IV destination databases. Do not use *Database Maintenance* (98.290.00) from a Solomon IV version 6.0 Client installation to create destination databases – this will cause *Transformation Wizard* to fail. *Solomon IV FastForward* - *Database Creation* (FF.290.00) can be run before, during, or after the *Integrity Wizard* process.

FastForward Database Creation Step-by-Step

> **Launch *Solomon IV FastForward - Database Creation* (FF.290.00) from the Solomon IV/FastForward 6.0 program group:**

1. *Solomon IV FastForward* - *Database Creation* (FF.290.00) appears. Enter the destination SQL Server name, the SA login ID, and the SA password in the appropriate fields and click **Connect**.



Destination SQL Server Name: FOH-HALFTIME

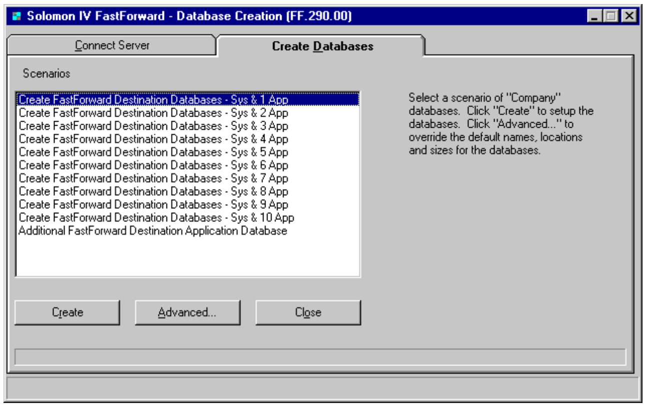
Login ID: sa

Password: xxxxxxxxxxxx

Connected to FOH-HALFTIME (Microsoft SQL Server 2000 - 8.00.760 (Intel X86)) as sa

Figure 44: Solomon IV FastForward - Database Creation (FF.290.00)

2. Once connected, click the **Create Databases** tab.



Create FastForward Destination Databases - Sys & 1 App
Create FastForward Destination Databases - Sys & 2 App
Create FastForward Destination Databases - Sys & 3 App
Create FastForward Destination Databases - Sys & 4 App
Create FastForward Destination Databases - Sys & 5 App
Create FastForward Destination Databases - Sys & 6 App
Create FastForward Destination Databases - Sys & 7 App
Create FastForward Destination Databases - Sys & 8 App
Create FastForward Destination Databases - Sys & 9 App
Create FastForward Destination Databases - Sys & 10 App
Additional FastForward Destination Application Database

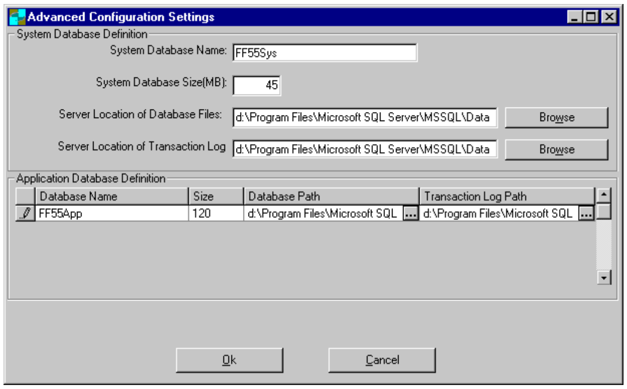
Select a scenario of "Company" databases. Click "Create" to setup the databases. Click "Advanced..." to override the default names, locations and sizes for the databases.

Figure 45: Solomon IV FastForward - Database Creation (FF.290.00), Create Databases tab

3. Select the appropriate choice from the list of scenarios. *Solomon IV FastForward* - *Database Creation* (FF.290.00) supports creating a system destination database along with up to 10 application destination databases. Select the appropriate scenario from the list and click **Advanced** to specify the database names, sizes, and physical file locations.

   Note: If it is necessary to create more than 10 application databases, you will need to use the Create FastForward Destination Databases - Sys & 10 App scenario to create the first 10 databases and the Additional FastForward Destination Database scenario to create each extra application database that is needed.

4. The *Advanced Configuration Settings* window displays. Change the **System Database Name** to the desired name. If desired, the **System Database Size**, **Server Location of Database Files**, and **Server Location of Transaction Log** can be changed as well. The grid contains the application database names, sizes, and paths. These fields can be changed as desired. After making the needed changes, click **Ok** to return to the **Create Databases** tab.



System Database Definition

System Database Name: FF55Sys

System Database Size(MB): 45

Server Location of Database Files: d:\Program Files\Microsoft SQL Server\MSSQL\Data

Server Location of Transaction Log: d:\Program Files\Microsoft SQL Server\MSSQL\Data

Application Database Definition

Database Name	Size	Database Path	Transaction Log Path
FF55App	120	d:\Program Files\Microsoft SQL	d:\Program Files\Microsoft SQL

Figure 46: Advanced Configuration Settings

Note: It is not necessary to determine what the exact size of the destination databases will be prior to creating the databases. *Solomon IV FastForward* **-** *Database Creation* (FF.290.00) creates the database such that they will automatically grow as needed.

5. In the **Create Databases** tab, click **Create** to begin creating the destination databases. After the process is complete, a confirmation window indicating the databases have been created successfully will be displayed. Click **OK** to close the window, then click **Close** to exit *Solomon IV FastForward* - *Database Creation* (FF.290.00).

Using Transformation Wizard

Transformation Wizard Overview

Once *Integrity Wizard* completes the database analysis with no Fatal Errors, the FastForward 6.0 *Transformation Wizard* can be run. This tool is designed to automate the launch of specially designed Microsoft Data Transformation Service (DTS) packages that pump accounting data from your old Solomon database to the new v6.0 structures. DTS is embedded functionality in the SQL Server 2000 suite and is therefore available automatically after SQL Server 2000 is installed.

Like *Integrity Wizard*, *Transformation Wizard* is written in Microsoft Visual Basic 6.0 and uses 32-bit Microsoft SQL Server and Pervasive.SQL 2.54 ODBC connections to link to the existing Solomon database. In addition, OLEDB database connection 85s are used to connect to the Solomon IV destination database. Once connected successfully, *Transformation Wizard* becomes a “data pump” and moves data rapidly between the old and new platforms. *Transformation Wizard* displays process information during execution to keep the user informed during the migration process.

Preliminary Migrations

The FastForward 6.0 tools include the ability to process a Preliminary Migration with *Transformation Wizard* in situations in which the database has not passed the integrity checks in *Integrity Wizard*. Although the Preliminary Migration will allow some of the integrity checks to be bypassed, error conditions such as invalid dates, null values and duplicate records do require correction prior to running *Transformation Wizard*. This functionality is not intended for bypassing Fatal Errors in order to complete “live” migrations, but rather to assist with testing of the migration process.

Performing a Preliminary Migration is also useful in situations where a migrated database is needed for client training and/or demonstration purposes using the client’s actual data. However, the Preliminary Migration should never be used as the final production database since there may be data irregularities that cause application or reporting issues.

Important: These databases will be marked to indicate that the migration was a Preliminary Migration. Microsoft Business Solutions will not provide support for production databases that were migrated using the Preliminary Migration functionality. A warning message is displayed on the Solomon IV menu in version 6.0 if the database is a Preliminary Migration database. Additionally, future versions of Solomon IV will prevent access to Preliminary Migration databases after 20 accesses.

Transformation Wizard Preprocess

Before the actual data migration begins, *Transformation Wizard* launches a preprocess that performs several tasks.

If upgrading from the Pervasive platform, intermediate databases are created on the destination SQL Server during the first step of the preprocess. The task is accomplished by calling Microsoft SQL-DMO routines via Solomon IV's SOLDB.DLL. The databases are named by appending "_Upg" to the end of the names of the destination system and application databases. This step is only performed when *Transformation Wizard* is running the upgrade on the database for the first time.

Intermediate database tables are created next, if necessary. In the case of an upgrade from the Pervasive platform where an intermediate database is required, the Solomon schema from the source database is mirrored in the intermediate database. Special tables to help facilitate the migration are also created. These tables are created via SQL statements that reside in the .CRT files in the SQL directory under *Transformation Wizard*. This step is also only performed when *Transformation Wizard* is running the upgrade on the database for the first time.

The remaining steps described below are performed every time an upgrade is run, regardless of whether the upgrade is being run for the first time or resumed from a previous attempt.

Next upgrade default tables are created in the intermediate database and populated. These tables hold default values provided by *Integrity Wizard* that populate various fields in the destination database. The file SOLUPTAB.CNV, located in the root of the FastForward 6.0 subdirectory, contains the schema and data values for the upgrade defaults.

Special processing for Preliminary Migrations takes place at this point. Preliminary Migrations are migrations that are launched without clearing all Fatal Errors identified by *Integrity Wizard* in the source database. The database is flagged so that it can be determined if an upgrade is a Live or a Preliminary Migration.

The next step creates the upgrade migration views. These views serve as the bridge between previous Solomon IV database schema and the current version of the Solomon IV database schema. These views are generated from SQL statements that reside in the .CRV files in the SQL directory under *Transformation Wizard*.

Some Solomon modules require external data to be merged with existing data in order to perform the upgrade successfully. The next step in the Preprocess stage handles importing any necessary data into temporary tables for joining during the upgrade. The PREPROCIMPORT.LST file in the SQL directory under *Transformation Wizard* lists the modules that need external data, the schema, database and table to import the data into as well as the data file to use for the import.

Indexes in the Solomon IV database may reduce database performance while the data is being moved to the destination databases, so the next preprocess step saves and removes indexes. The indexes are restored after the upgrade is complete. Note that, in order to preserve data integrity in the database, no unique indexes are removed.

Next, all log files are purged from the Logs directory under *Transformation Wizard* if the **Purge upgrade log files** checkbox was marked on *Confirm Transformation* (TW.040.00).

The last task performed during the preprocessing stage is to count the number of steps it will take to complete the current upgrade. The DTS packages are each loaded and the steps in each one are counted and accumulated. This information allows *Transformation Wizard* to provide status and progress feedback during the data transformation process.

Preprocess Summary

1. Create intermediate databases (if necessary)
2. Generate upgrade tables
3. Generate and populate upgrade default values tables
4. Mark database as Preliminary Migration (if necessary)
5. Generate migration views
6. Data import
7. Save and drop destination database indexes
8. Purge log files (if chosen)
9. Count upgrade steps

Transformation Wizard Postprocess

The following is a description of the tasks that are performed during the postprocess stage of *Transformation Wizard* during an upgrade. When the upgrade is a Preliminary Migration, the *Transformation Wizard* finishes processing and marking the database as Preliminary Migration. The next task is restoring the database indexes that were dropped during the preprocess.

Then postprocess scripts are executed against the destination databases. The scripts that are executed can be found in the SQL directory under *Transformation Wizard* and are listed in the POSTPROCSCRIPTS.LST file. The scripts are executed only if: (1) the data transformation portion of the upgrade completes successfully and (2) the module for which the script is being run was part of the upgrade (as listed in POSTPROCSCRIPTS.LST).

Lastly, if the upgrade is from the Pervasive database platform and was completed successfully, the intermediate databases are removed from the SQL Server.

Postprocess Summary

1. Finish marking database as Preliminary Migration (if necessary)
2. Restore dropped indexes to destination database
3. Run postprocess scripts
4. Remove intermediate databases (if completed successfully)

CNV File Descriptions

FastForward 6.0 uses two *.CNV files: SOLUPGRA.CNV and SOLUPTAB.CNV. These files contain information that is shared between the FastForward *Integrity Wizard* and *Transformation Wizard*. The file formats are similar to standard Microsoft Windows *.INI files.

SOLUPGRA.CNV contains most of the information shared between the FastForward 6.0 tools. This information includes certain upgrade parameters, file paths, database default values and database connection information. Also contained in SOLUPGRA.CNV is information regarding the modules that have been chosen to upgrade and table data for each of the tables being migrated from the source database.

SOLUPTAB.CNV is a more specialized file. It contains table definitions and values for the upgrade default value tables. This data is used to dynamically create default value tables during the preprocess stage of the *Transformation Wizard* upgrade process.

Data Migration

Once the preprocess completes, the actual data migration process can be launched. For Solomon IV version 2.06, *Transformation Wizard* runs a two-step process. During the first step, *Transformation Wizard* moves data from the Pervasive source database to an exact replica in SQL Server.

This part of the migration uses specially designed DTS packages. Package names are assigned according to the step they represent for a module during a migration. The names reflect the Solomon module, source schema and destination schema versions. Upgrades from the Pervasive platform require two steps per module, so two DTS packages are required. Upgrades from SQL Server require only one step, and thus one package.

Package names begin with several characters representing the Solomon module that the package upgrades. The next two characters represent the source schema version, followed by a T (for “To”), followed by two characters that represent the destination schema version.

For example, these are the file names for the General Ledger migration DTS packages:

FMGGL20T20 .DTS	Step 1 of 2, copy data from source database to interim database. This step copies data from the source Solomon IV 2.06 database to an interim database on the destination SQL Server that mirrors the Solomon IV 2.06 schema.
FMGGL20T60 .DTS	Step 2 of 2, convert migrate data from the interim database to destination database. This step converts the data from the interim Solomon IV 2.06 database to the

	Solomon IV destination database.

Occasionally, there may be database conditions that cannot be identified by *Integrity Wizard* and therefore result in errors during *Transformation Wizard*. These are typically irregularities that involve invalid date values, null values or duplicate records that require research to resolve. *Transformation Wizard* will indicate the tables that encountered errors in the Logs located in the FastForward\TransformationWizard\Logs directory.

Transformation Wizard Log Files

Transformation Wizard generates several log files during the course of an upgrade. The most important of these log files are the database table names prefixed by DTS_ or TW_ or the DTS step name prefixed by Pkg_. The log files can be found in the Logs subdirectory under the *Transformation Wizard* subdirectory.

The TW_ and DTS_ logs can be viewed with the log viewer built into *Transformation Wizard* by double-clicking on the table grid on the final screen *Transformation Wizard* (TW.300.00). Pkg_ logs must be manually loaded in Notepad or some other text editor in order to view them.

Logs that begin with Pkg_ are automatically generated by Microsoft's DTS service when *Transformation Wizard* executes each DTS package. These logs list basic information about the DTS package, each step that was executed, and whether that step passed or failed. Each step in the DTS package correlates with one table being upgraded in that package. Generally the step names are cryptic, so in order to find out what table they correspond to you may have to cross-reference with the step names listed in the DTS_ logs files or open up the DTS package through SQL Server's Enterprise Manager.

Logs that begin with DTS_ are also automatically generated by Microsoft's DTS service when data is being transformed from the source database to the destination database. The name of the file following the DTS_ corresponds with the name of the table that the log file pertains to. These files are generated for every table that is upgraded by *Transformation Wizard*.

Logs that begin with TW_ are generated from information gathered internally by *Transformation Wizard* itself. Like the DTS_ logs, the name of the log following the TW_ corresponds to the table that the log file is for. These logs are generated only when an error occurs for a particular table being upgraded. In some instances *Transformation Wizard* is able to gather more data on errors that occur with Microsoft DTS than what DTS automatically provides in its log files. The TW_ log files are provided as an additional tool to help debug problems, and in some cases are more helpful than the DTS_ logs.

When researching errors that occur during a migration, start with the package name (filename) and the database table name in which the error occurred.

Log File Summary

Pkg_ Logs

Generated by DTS for each package executed. Information provided:

- Package Name — typically the filename of the package
- Package Description — description, if present
- Execution Info — computer name, computer login, start and completion times
- Package Step Statuses — a line written for each step (table) and its result

DTS_ Logs

Generated by DTS for each step (table) completed. Information provided:

- Package Name — package where step resides
- Package Description — description, if present
- Step Name — name of step, often not descriptive
- Execution Info — start and completion times
- Error Info — error messages or rows of data where errors occurred

TW_ Logs

Generated by *Transformation Wizard* only when an error occurs while migrating a table. Corresponds to a DTS_ log for the same table. Information provided:

- Time — time error was logged
- Source — where error originated within DTS
- Table — table where error occurred
- Error — error number and message returned by DTS

Transformation Wizard Preparation Checklist

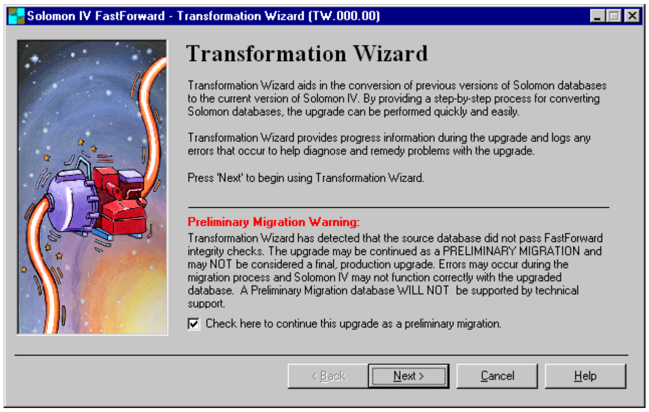
> **Review this checklist prior to running *Transformation Wizard*. Addressing these items prior to running *Transformation Wizard* expedites the migration process.**

Transformation Wizard Preparation Checklist		
X		**Action Item**
	1.	Verify Solomon IV version 6.0 is installed.
	2.	Verify a restorable backup of the Solomon IV 2.06 source database is available.
	3.	Verify that there is enough disk space to accommodate the migrated database.
	4.	Create destination databases using *FastForward 6.0 - Database Creation* (FF.290.00).
	5.	Verify all Fatal Error conditions identified by *Integrity Wizard* are successfully resolved by running a final pass through *Integrity Wizard*.

Transformation Wizard Step-by-Step

> **Launch FastForward *Transformation Wizard* from the Solomon IV program group:**

1. The *Transformation Wizard* (TW.000.00) introductory screen appears. Review the introduction and click **Next** to proceed. If you are performing a preliminary (or "test") migration, click on the **Check here to continue...** checkbox at the bottom of the screen. Note that this is a necessary step if the *Integrity Wizard* found Fatal Errors which were not corrected prior to running the Transformation Wizard.



Transformation Wizard

Transformation Wizard aids in the conversion of previous versions of Solomon databases to the current version of Solomon IV. By providing a step-by-step process for converting Solomon databases, the upgrade can be performed quickly and easily.

Transformation Wizard provides progress information during the upgrade and logs any errors that occur to help diagnose and remedy problems with the upgrade.

Press 'Next' to begin using Transformation Wizard.

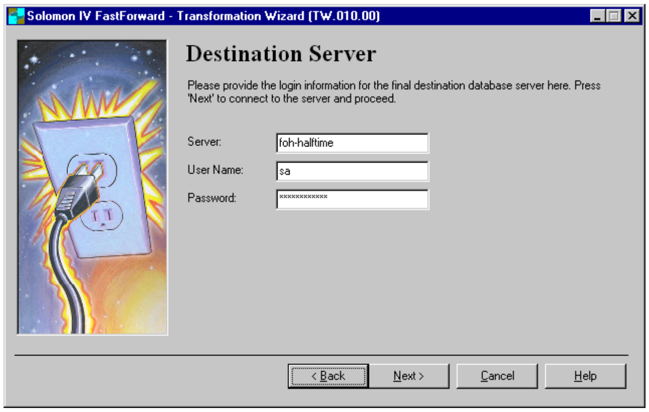
Preliminary Migration Warning:

Transformation Wizard has detected that the source database did not pass FastForward integrity checks. The upgrade may be continued as a PRELIMINARY MIGRATION and may NOT be considered a final, production upgrade. Errors may occur during the migration process and Solomon IV may not function correctly with the upgraded database. A Preliminary Migration database WILL NOT be supported by technical support.

☑ Check here to continue this upgrade as a preliminary migration.

Figure 47: Transformation Wizard (TW.000.00) — Preliminary Migration Warning

2. *Destination Server* appears next. Enter the **Server**, **User Name** and **Password** if applicable. Then click **Next** to continue.



Solomon IV FastForward - Transformation Wizard (TW.010.00)

Destination Server

Please provide the login information for the final destination database server here. Press 'Next' to connect to the server and proceed.

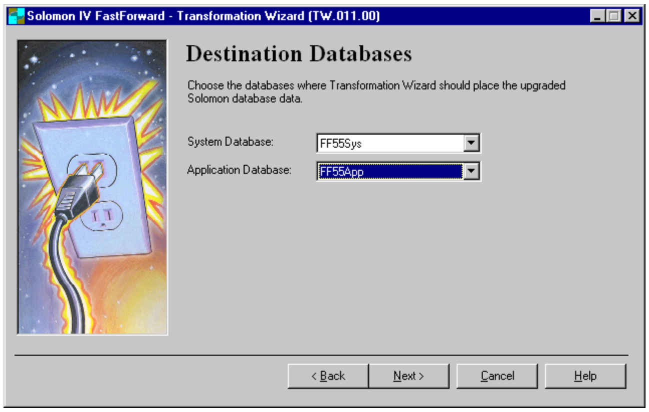
Server: foh-halftime

User Name: sa

Password: ************

Figure 48: Transformation Wizard (TW.010.00) — Destination Server

3. *Destination Databases* appears next. Select the SQL Server **System Database** and **Application Database** from the drop-down list. Then click **Next** to continue.



Destination Databases

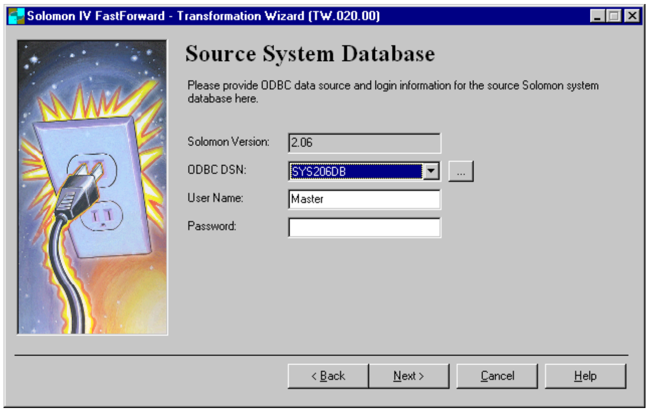
Choose the databases where Transformation Wizard should place the upgraded Solomon database data.

System Database: FF55Sys

Application Database: FF55App

Figure 49: Transformation Wizard (TW.011.00) — Destination Databases

4. *Source System Database* connection information is requested.



Source System Database

Please provide ODBC data source and login information for the source Solomon system database here.

Solomon Version: 2.06

ODBC DSN: SYS206DB

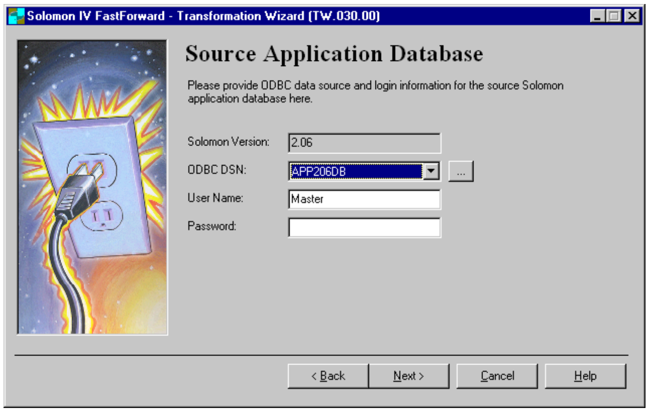
User Name: Master

Password:

Figure 50: Transformation Wizard (TW.020.00) — Source System Database

- **Solomon Version** displays the version of the source database being migrated. This information is obtained from the file SOLUPGRA.CNV that is created by *Integrity Wizard*.
- Verify the **ODBC DSN** for the source system database. Modify if necessary by making a selection from the drop-down list or click the **Ellipsis** button to launch the standard ODBC Data Source Administrator dialog to select or create a new data source name for the source system database.
- Specify the **User Name** and **Password** for the source system database.
- Click **Next** to continue.

5. Complete the *Source Application Database* login information.



Solomon IV FastForward - Transformation Wizard (TW.030.00)

Source Application Database

Please provide ODBC data source and login information for the source Solomon application database here.

Solomon Version: 2.06

ODBC DSN: APP206DB

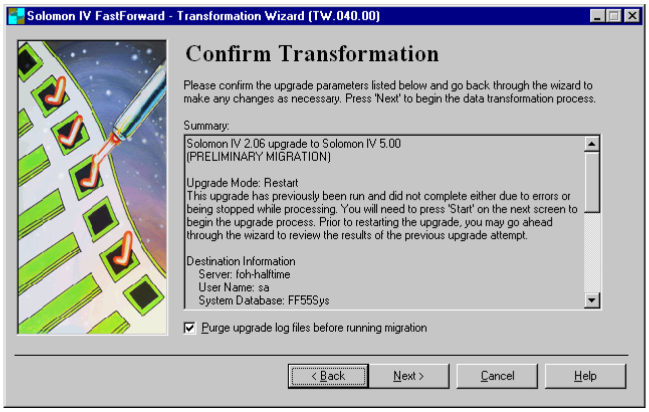
User Name: Master

Password:

Figure 51: Transformation Wizard (TW.030.00) — Source Application Database

- **Solomon Version** displays the version of the source database being migrated. This information is obtained from the file SOLUPGRA.CNV that is created by *Integrity Wizard*.
- Verify the **ODBC DSN** for the source application database. Modify if necessary by making a selection from the drop-down list or click the **Ellipsis** button to launch the standard Microsoft Data Link Properties dialog to select or create a new data source name for the source application database.
- Specify the **User Name** and **Password** for the source application database.
- Click **Next** to continue.

6. *Confirm Transformation* appears listing the upgrade parameters defined for the migration for review.



Confirm Transformation

Please confirm the upgrade parameters listed below and go back through the wizard to make any changes as necessary. Press 'Next' to begin the data transformation process.

Summary:

Solomon IV 2.06 upgrade to Solomon IV 5.00
(PRELIMINARY MIGRATION)

Upgrade Mode: Restart
This upgrade has previously been run and did not complete either due to errors or being stopped while processing. You will need to press 'Start' on the next screen to begin the upgrade process. Prior to restarting the upgrade, you may go ahead through the wizard to review the results of the previous upgrade attempt.

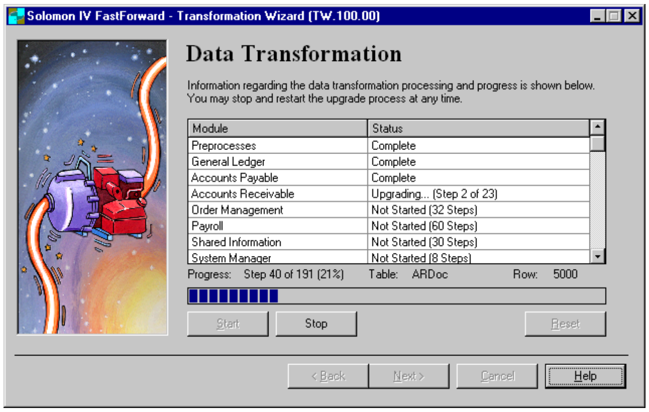
Destination Information
Server: foh-halftime
User Name: sa
System Database: FF55Sys

☑ Purge upgrade log files before running migration

Figure 52: Transformation Wizard (TW.040.00) — Confirm Transformation

- Select **Purge upgrade log files before running migration** to delete all log files created in previous passes through *Transformation Wizard*.
- If the upgrade parameters are correct, click **Next** to proceed. If modifications are required, click **Back** to make corrections in the previous screens or **Cancel** to exit *Transformation Wizard*.

7. *Data Transformation* appears and the upgrade process automatically launches.



Data Transformation

Information regarding the data transformation processing and progress is shown below.
You may stop and restart the upgrade process at any time.

Module	Status
Preprocesses	Complete
General Ledger	Complete
Accounts Payable	Complete
Accounts Receivable	Upgrading... (Step 2 of 23)
Order Management	Not Started (32 Steps)
Payroll	Not Started (60 Steps)
Shared Information	Not Started (30 Steps)
System Manager	Not Started (8 Steps)

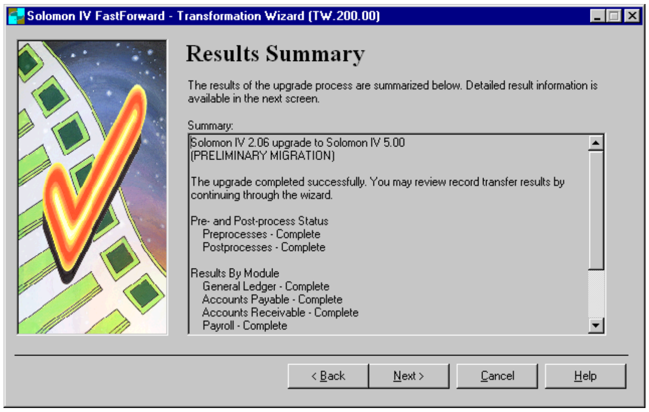
Progress: Step 40 of 191 (21%) Table: ARDoc Row: 5000

Figure 53: Transformation Wizard (TW.100.00) — Data Transformation

- As the migration process proceeds, the **Status** of each module is updated to reflect the result of the migration. Complete indicates that the migration for the module was successful. Error indicates that the *Transformation Wizard* encountered conditions that prevented the module from upgrading successfully. Details regarding errors encountered are provided in the DTS and TW log files. The log files are located in the Logs subdirectory under the *Transformation Wizard* directory. In addition, the log files can be directly accessed through the *Transformation Wizard* (TW.300.00) — *Results Details*.
- Click **Stop** to pause the migration process and **Start** to restart.

- Click **Reset** to delete migrated data.
- Click **Next** to continue.

8. *Results Summary* displays summary level results of the migration, including any tables that did not migrate successfully and the names of the DTS and TW error logs associated with these tables. Click **Next** to review detailed results.



Results Summary

The results of the upgrade process are summarized below. Detailed result information is available in the next screen.

Summary:

Solomon IV 2.06 upgrade to Solomon IV 5.00
(PRELIMINARY MIGRATION)

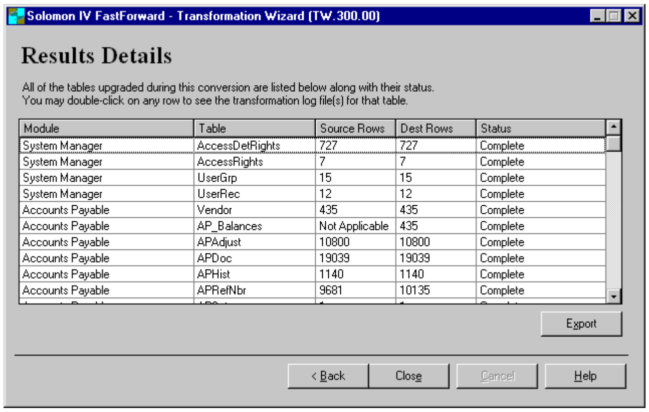
The upgrade completed successfully. You may review record transfer results by continuing through the wizard.

Pre- and Post-process Status
   Preprocesses - Complete
   Postprocesses - Complete

Results By Module
   General Ledger - Complete
   Accounts Payable - Complete
   Accounts Receivable - Complete
   Payroll - Complete

Figure 54: Transformation Wizard (TW.200.00) — Results Summary

9. *Results Details* displays next and lists all upgraded modules and corresponding tables including record counts for the source and destination database as well as the migration status for the each table.



Results Details

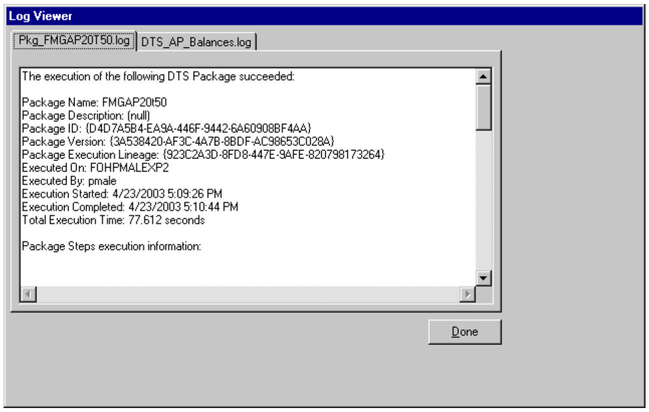
All of the tables upgraded during this conversion are listed below along with their status.
You may double-click on any row to see the transformation log file(s) for that table.

Module	Table	Source Rows	Dest Rows	Status
System Manager	AccessDetRights	727	727	Complete
System Manager	AccessRights	7	7	Complete
System Manager	UserGrp	15	15	Complete
System Manager	UserRec	12	12	Complete
Accounts Payable	Vendor	435	435	Complete
Accounts Payable	AP_Balances	Not Applicable	435	Complete
Accounts Payable	APAdjust	10800	10800	Complete
Accounts Payable	APDoc	19039	19039	Complete
Accounts Payable	APHist	1140	1140	Complete
Accounts Payable	APRefNbr	9681	10135	Complete

Figure 55: Transformation Wizard (TW.300.00) — Results Details

Important: Some tables are likely to have variances in record counts. Review "Record Count Variances" for details on these tables.

10. Double-click on any row to display the transformation log file for the selected table.

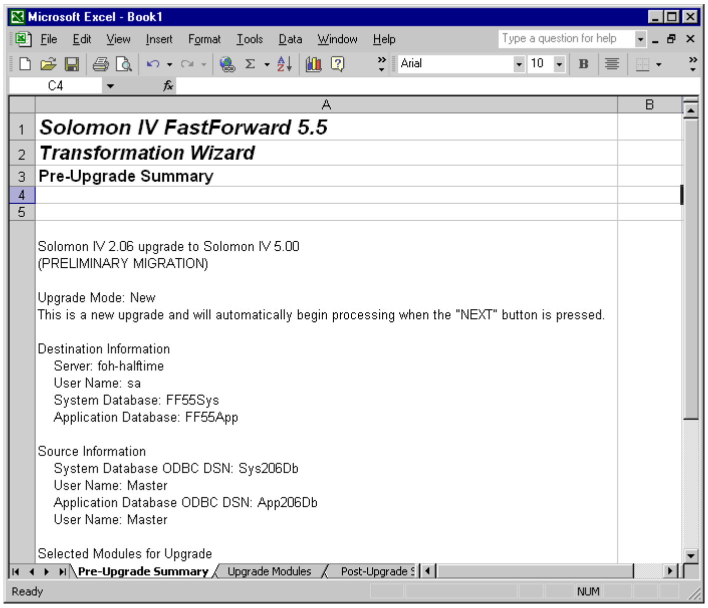


```
The execution of the following DTS Package succeeded:

Package Name: FMGAP20t50
Package Description: (null)
Package ID: {D4D7A5B4-EA9A-446F-9442-6A60908BF4AA}
Package Version: {3A538420-AF3C-4A7B-8BDF-AC98653C028A}
Package Execution Lineage: {923C2A3D-8FD8-447E-9AFE-820798173264}
Executed On: FOHPMALEXP2
Executed By: pmale
Execution Started: 4/23/2003 5:09:26 PM
Execution Completed: 4/23/2003 5:10:44 PM
Total Execution Time: 77.612 seconds

Package Steps execution information:
```

Figure 56: Log Viewer

11. DTS, PKG, and TW logs will be available in the viewer in the event that an error was encountered on the selected table.

12. Click **Export** in *Transformation Wizard* (TW.300.00) — *Results Details* to create an Excel spreadsheet that includes four separate worksheets: Pre-Upgrade Summary, Upgrade Modules, Post-Upgrade Summary, and Upgrade Tables.



Solomon IV FastForward 5.5

Transformation Wizard

Pre-Upgrade Summary

Solomon IV 2.06 upgrade to Solomon IV 5.00
(PRELIMINARY MIGRATION)

Upgrade Mode: New
This is a new upgrade and will automatically begin processing when the "NEXT" button is pressed.

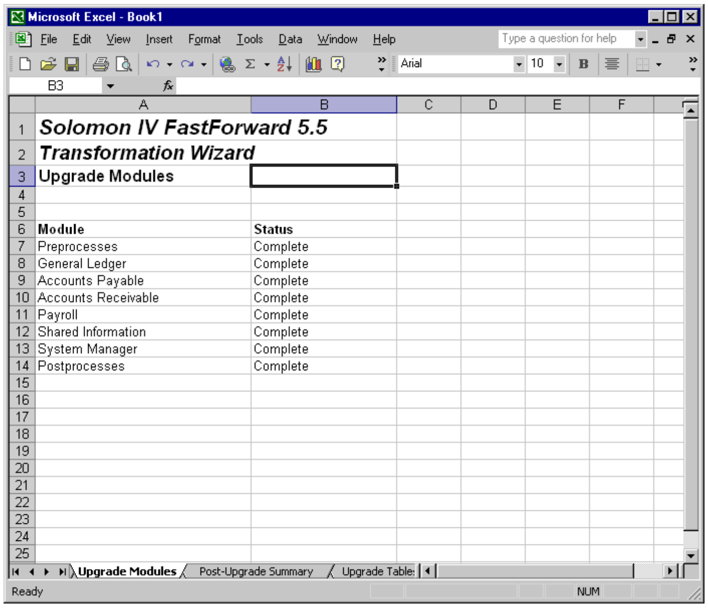
Destination Information
Server: foh-halftime
User Name: sa
System Database: FF55Sys
Application Database: FF55App

Source Information
System Database ODBC DSN: Sys206Db
User Name: Master
Application Database ODBC DSN: App206Db
User Name: Master

Selected Modules for Upgrade

Figure 57: Pre-Upgrade Summary

13. Pre-Upgrade Summary lists all information that was provided in *Transformation Wizard* (TW.040.00) — *Confirm Transformation*.



Solomon IV FastForward 5.5

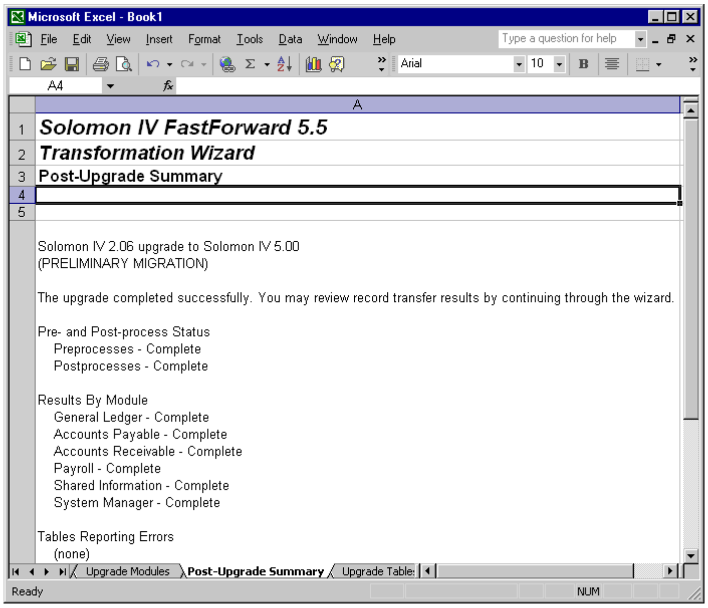
Transformation Wizard

Upgrade Modules

Module	Status
Preprocesses	Complete
General Ledger	Complete
Accounts Payable	Complete
Accounts Receivable	Complete
Payroll	Complete
Shared Information	Complete
System Manager	Complete
Postprocesses	Complete

Figure 58: Upgrade Modules

14. Upgrade Modules lists each module that was migrated and the status of the migration for the module.



Solomon IV FastForward 5.5

Transformation Wizard

Post-Upgrade Summary

Solomon IV 2.06 upgrade to Solomon IV 5.00
(PRELIMINARY MIGRATION)

The upgrade completed successfully. You may review record transfer results by continuing through the wizard.

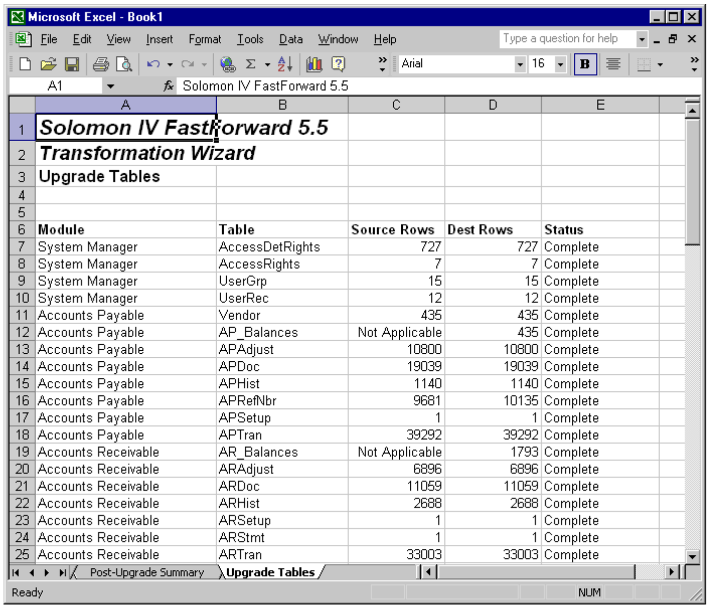
Pre- and Post-process Status
Preprocesses - Complete
Postprocesses - Complete

Results By Module
General Ledger - Complete
Accounts Payable - Complete
Accounts Receivable - Complete
Payroll - Complete
Shared Information - Complete
System Manager - Complete

Tables Reporting Errors
(none)

Figure 59: Post-Upgrade Summary

15. Post-Upgrade Summary also includes information regarding the modules that migrated, but also includes tables that reported errors during the migration process.



Solomon IV FastForward 5.5

Solomon IV FastForward 5.5

Transformation Wizard

Upgrade Tables

Module	Table	Source Rows	Dest Rows	Status
System Manager	AccessDetRights	727	727	Complete
System Manager	AccessRights	7	7	Complete
System Manager	UserGrp	15	15	Complete
System Manager	UserRec	12	12	Complete
Accounts Payable	Vendor	435	435	Complete
Accounts Payable	AP_Balances	Not Applicable	435	Complete
Accounts Payable	APAdjust	10800	10800	Complete
Accounts Payable	APDoc	19039	19039	Complete
Accounts Payable	APHist	1140	1140	Complete
Accounts Payable	APRefNbr	9681	10135	Complete
Accounts Payable	APSetup	1	1	Complete
Accounts Payable	APTran	39292	39292	Complete
Accounts Receivable	AR_Balances	Not Applicable	1793	Complete
Accounts Receivable	ARAdjust	6896	6896	Complete
Accounts Receivable	ARDoc	11059	11059	Complete
Accounts Receivable	ARHist	2688	2688	Complete
Accounts Receivable	ARSetup	1	1	Complete
Accounts Receivable	ARStmt	1	1	Complete
Accounts Receivable	ARTran	33003	33003	Complete

Figure 60: Upgrade Tables

16. Upgrade Tables lists each table that was migrated, the number of records in the source and destination database and the status of the migration for the table.

Record Count Variances

The following tables may have variances in record counts between the Solomon IV 2.06 source and Solomon IV 6.0 destination databases:

- **ACCESSRIGHTS** — The record count of the destination Solomon IV 6.0 database should be the same as that of the source Solomon IV 2.06 database except when records must be dropped to pass *Integrity Wizard*.
- **ACCESSDETRIGHTS** — The record count of the destination Solomon IV 6.0 database should be the same as that of the source Solomon IV 2.06 database except when records must be dropped to pass *Integrity Wizard*.
- **ACCTHIST** — The record count of the destination Solomon IV 6.0 database should be the same as that of the source Solomon IV 2.06 database unless the client is maintaining budget information, in which case the record count will increase. In previous versions, budget information was stored in the same record within AcctHist. In Solomon IV v6.0 budget information is kept in separate AcctHist records. When Currency Manager is being used, the record count should be lower. In prior versions of Solomon, foreign currency balances were stored in ACCTHIST. These balances have been moved into the CURYACCT table, causing the number of records in ACCTHIST to be lower in Solomon IV v6.0.

- **CURYACCT** — The record count of ACCTHIST in the destination Solomon IV 6.0 database should be the same as that of the source Solomon IV 2.06 database when Currency Manager is not used since CURYACCT is populated from the ACCTHIST table in this instance. When Currency Manager is used, the record count should be higher after migration. In Solomon IV version 6.0, CURYACCT contains both base currency and foreign currency balances with base currency translations.
- **INUNIT** — The record count of the destination Solomon IV 6.0 database should be higher than that of the source Solomon IV 2.06 database because entries must be made for 1-to-1 migrations for stock units.
- **LOCTABLE** — The record count of the destination Solomon IV 6.0 database will likely be lower than that of the source Solomon IV 2.06 database because the index for LOCTABLE in Solomon IV 2.06 includes SITEID, WHSELOC and INVTID. In Solomon IV v6.0 the index is only SITEID and WHSELOC.
- **POTRAN** — The record count of the destination Solomon IV 6.0 database may be higher than that of the source Solomon IV 2.06 database because freight information has moved from the header to the detail in Solomon IV v6.0.
- **PURORDDET** — The record count of the destination Solomon IV 6.0 database may be higher than that of the source Solomon IV 2.06 database because freight information has moved from the header to the detail in Solomon IV v6.0.

Post-Migration Tasks

Post-Migration Overview

Once *Transformation Wizard* finishes, the following tasks should be considered to complete the upgrade:

- **Validate Database Migration** — Login to the migrated database and print the reports listed in the *Source Database Reports Checklist* (page 55) and compare to the reports printed from the Solomon IV 2.06 database.
- **Messages, PVRec and RptControl** — Run DBUpdate on messages.msg, pvrec.csv and rptcontrol.csv, all found in the Sol4\DB\CSV folder, to update those tables in the migrated database.
- **Access Rights** — Review and update *Access Rights Maintenance* (95.270.00) in the migrated database. Several new screens are available in Solomon IV v6.0 and security will need to be established for these screens.

  Important: If a single system database was used to migrate multiple application databases, only the access rights for the last database migrated will be available in the migrated system database. A post-process in *Transformation Wizard* populates the COMPANY, INTERCOMPANY, SUBXREF, and ACCTXREF tables. Running the migration again, using the same destination system database will not truncate these tables. The resulting issue is that SUBXREF and ACCTXREF will not be populated correctly and access rights will only apply to the last database migrated.

To correct SUBXREF and ACCTXREF, log into Solomon IV version 6.0 and create and save a new account in General Ledger’s *Chart of Accounts Maintenance* (01.260.00) and a new subaccount in *Subaccount Maintenance* (01.270.00). Delete the newly created account and subaccount. This triggers the stored procedure that will update SUBXREF and ACCTXREF.

Access rights will need to be reviewed and updated.

- **Registration** – Register Solomon IV version 6.0 by entering the module registration information, including unlocking keys, in the REGISTRATION screen on Solomon IV’s Utility menu.
- **Install Database at client site** — If the migration was done offsite, the process to install the database at the client site is as follows:

  1. Backup the database using SQL Server’s BACKUP functionality.
  2. Copy the database to a zip drive or other appropriate media.
  3. At the client site verify that SQL Server 2000 and Solomon IV v6.0 is installed.
  4. Use SQL Server’s RESTORE functionality to restore the database.

Product Updates

Product Updates Overview

Updates to FastForward version 6.0 should be applied regularly. As they are created, the most current FF6.0 updates will be available for download at:

ftp://ftp.solomon.com/s4/FastForward/FFUpdates/FF6.0/ff60_update.exe

Download the ff60_update.exe file and execute after FastForward 6.0 has been installed. You must enter your FastForward 6.0 installation path for the updates to apply correctly. This process must be done for each update that is downloaded.

Note: If this location does not contain a ff60_update.exe file, then updates have yet to be created for FastForward 6.0.

Upgrading FRx

FRx Upgrade Overview

Solomon IV version 6.0 uses FRx® version 6.7. Upgrading from FRx 4.7 to FRx version 6.7 is performed separately from the FastForward 6.0 migration. Instructions for upgrading to FRx version 6.7 are available at the following location:

FRx 4.7 to 6.7 Conversion Guide.pdf

Index