Installation Guide
QAD Configurator

78-0914A
QAD Configurator 5.2.1
September 2010

Configurator_IG_v521.pdf/hes/hes

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

Contents

Chapter 1 Installing QAD Configurator 1

Overview 2
Software Prerequisites 2
Deployment and Hardware Requirements 3
Configurator Installation Preparations 5
Installation Steps 6
Install QAD Configurator Files 7
Create QAD Configurator Databases (Progress Database) 9
Create QAD Configurator Databases (Oracle Database) 14
Create Start/Stop Scripts for the Configurator Production Databases . . 18
Modify QAD Configurator AppServer API File 19
Configure the Site-Specific Data Creation Feature 20
Compile QAD Configurator AppServer API Files (Progress Database) 20
Compile QAD Configurator AppServer API Files (Oracle Database) . 22
Modify the QAD .NET UI .pf File (Progress Database) 24
Modify the QAD .NET UI .pf File (Oracle Database) 25
Modify the QAD .NET UI AppServer Configuration 25
Modify WebSpeed Configurations 26
Set Up QAD Configurator .NET UI Plug-In 27
Set Up QAD Configurator Process Maps 27
Register QAD Configurator 28
Configure WebSpeed Settings 28
Modify Desktop Telnet Scripts 29
Set QAD Desktop Connection Timeout (Optional) 30
Integrate with Trade Management (Optional) 30

Chapter 2 Upgrading QAD Configurator 35

Overview 36
Prior to Upgrading QAD Configurator 36
Upgrading Steps 36
Installing QAD Configurator 5.2.1 36
Copying Data From the Previous Version of QAD Configurator 37
Updating Database Schema 37
Converting Production Database to UTF-8 38

Converting Production Database to the Current Version and Loading Production Data 38
Rebuilding Database Index 39
Converting Administration Database 39
Converting Production Database to Enterprise Edition 39
Modifying the manifest.qpkg File 39

Appendix A Configurator Web Service APIs 41
Configurator Web Service APIs Overview 42
Session Context 42
Exceptions 45
Configurator APIs 45
Retrieve Configuration Groups 46
Create a New Variant Item 46
Retrieve Configurations 47
Find or Create Variant Item 49

Appendix B Troubleshooting 51

Index 55

Chapter 1

Installing QAD Configurator

Software Prerequisites 2

Configurator Installation Preparations 5

Installation Steps 6

Overview

QAD Configurator can be deployed on both Progress and Oracle databases, depending on the existing database environment of your QAD Enterprise Applications implementation. This chapter provides installation instructions for both these two deployment scenarios. Most steps are identical for Progress and Oracle implementations. Where steps differ, follow the instructions to use database-specific steps.

Software Prerequisites

The following components must be installed before you install QAD Configurator. See related documentation for information on how to install these components.

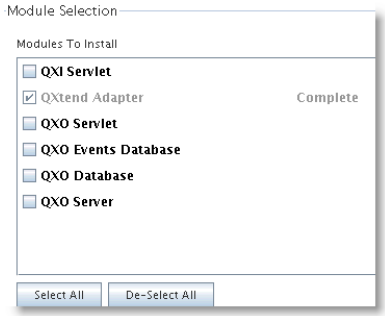
- OpenEdge 10.1A02 or later with at least the following components:

  Note If you want to use the site-specific data creation feature, you must install OpenEdge 10.1C01 or later.

  - OE Application Server Basic or OE Application Server Enterprise
  - OE Enterprise RDBMS
  - 4GL Development System
- One of the following supported versions of QAD Enterprise Applications:
  - MFG/PRO eB2.1 SP4
  - QAD 2007 and QAD 2007.1
  - QAD 2008 SE (Standard Edition)
  - QAD 2008.1 SE
  - QAD 2008.1 EE (Enterprise Edition)
  - QAD 2009 SE
  - QAD 2009 EE
  - QAD 2009.1 EE
  - QAD 2010 EE
  - QAD 2010.1 EE
- Apache Web server 2.0 or later

  Note QAD Configurator Web applications cannot be deployed on the Tomcat server.
- QXtend Adapter 1.6.2 or later

Fig. 1.1
QXtend Adapter Module Selection



Installing QAD Configurator on an Oracle database also requires the following components:

- OpenEdge 10.1C01 or 10.1C02 with Oracle DataServer
- Oracle Version 10gR1 (10.1.0.4.0) or 10gR2 (complete installation)

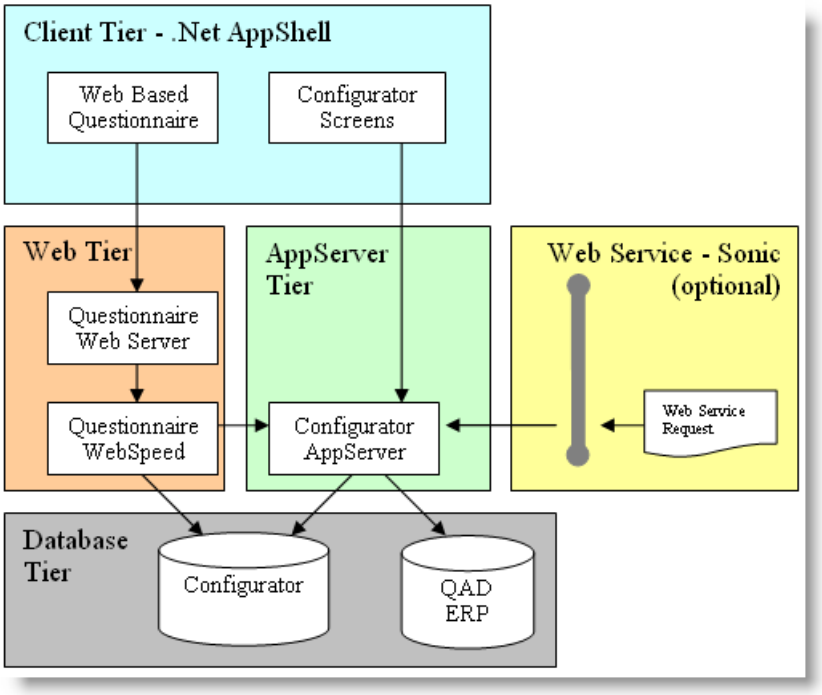
Deployment and Hardware Requirements

Depending on how you installed QAD .NET UI, you could have a single-tier or multi-tier QAD .NET UI environment. Either way, the default deployment of QAD Configurator places QAD Configurator databases on the QAD Enterprise Applications database server and its application files on the QAD .NET AppServer.

To implement the Questionnaire .NET UI—one of Configurator's components—install Configurator AppServer API and Configurator .NET plug-in files on the QAD .NET UI AppServer and install Configurator WebSpeed files on the Apache Web server.

Note QAD Configurator Web applications cannot be deployed on the Tomcat server.

Fig. 1.2
QAD Configurator Deployment Architecture



Client Tier (QAD .NET UI)

Configurator has been rewritten to run as native Windows screens run within the QAD .NET UI. Configurator Questionnaire is also launched within the .NET UI but uses Web-based technologies (HTML, JavaScript and AJAX) in order to be compatible with other Web-based applications, such as QAD CSS. Regarding sizing of .NET clients, see *Deployment Guide: QAD .NET User Interface* for the recommended requirements specified for the .NET UI.

Web Tier

The Web tier is responsible for managing the Web-based questionnaire. Depending on the volume of traffic that goes through the questionnaire this tier can either have a dedicated server or be combined with the AppServer tier.

Since this tier is simply responding to users' Web requests, with business logic executing on the AppServer tier, if deployed on a separate server, there should not be a lot of processing, disk, or memory requirements on this server. A low-end server consisting of a single dual-core CPU with 2 GB of memory and 60 GB hard drive should be sufficient.

AppServer Tier

The AppServer tier executes the business logic for the Configurator. It is responsible for handling Questionnaire business logic, Configurator screen business logic, and Web service API processing, if the environment is configured for Web service APIs.

Generally, the use of the Configurator will focus on the Questionnaire, as the Configurator .NET screens are used intermittently to set up and maintain the configurations. Typically the Web services would not be enabled. In this scenario two dual-core CPU with 4 GB of memory and 60 GB hard disk should suffice.

As mentioned previously, this server can be shared with the Web tier if the use of the Questionnaire does not create excess load on this server; otherwise, the Web tier should be separated to its own server.

Database Tier

The database tier hosts the database servers used to access the Configurator and QAD Enterprise Applications databases. Since QAD Enterprise Applications databases will typically have been already deployed to an optimal server, it is recommended that the Configurator database server also be deployed on the same server. If there are concerns about overloading the existing QAD Enterprise Applications database server, then a separate server can be configured for this database or the database server could be configured on the AppServer machine. If a separate server is desired, it is recommended that a two dual-core CPU machine with 4 GB of memory and 100 GB hard drive be used for the Configurator database server.

Web Service Tier

The optional Web service tier is used to host Sonic, the component that exposes Configurator APIs as Web services. This server does very little in that it provides the Web server used to host the Web services and routes the requests through the Configurator AppServer. If there is not excessive use of these Web services, Sonic can be deployed on the AppServer server. If there is extensive use of Web service calls, a dedicated server with a single dual-core CPU, 2 GB of memory and 60 GB hard drive should be used.

Configurator Installation Preparations

We recommend you plan your Configurator installation and collect the following information or environment parameters in a worksheet prior to installing Configurator. Replace the examples with your working environment parameter values. You will need this information during the installation process.

Table 1.1
Configurator Installation Worksheet

Prerequisite Information	Variable	Example
Progress installation directory	*ProgressInstallDir*	/dr01/progress/dlc101c
QAD Enterprise Applications database server installation directory	*QADERPInstallDir*	/dr01/qad/qad2009ee
Tomcat server installation directory	*TomcatInstallDir*	/dr01/tomcat
QAD .NET UI configuration name that identifies the environment in which you want to install Configurator. Desktop UI systems are typically built for multiple environments: pilot, production, training, development, and so forth. Each QAD UI system (Desktop-only, or combined Desktop and QAD .NET UI) has a unique name, which is reflected in the Tomcat directory structures: *TomcatInstallDir*/webapps/*qaduiConfig*	*qaduiConfig*	QADUIDemo
QAD .NET UI (AppShell) version. You can find this information in the `version.net` file under the Tomcat server installation directory.		2.8.2
QAD .NET UI AppServer Service Name. You can find this information by selecting Help|View Configuration from the QAD Enterprise Applications .NET UI.	*qaduiASService*	qadui_ASqaddemo

Prerequisite Information	Variable	Example
QAD .NET UI .pf file. The location of this file can be found in the `ubroker.properties` file under the *`ProgressInstall`*`/Properties` directory. Under the [UBroker.AS.*`qadui ASService`*] section in the file, you can find the full path to `base-live-set.pf` specified after -pf as a part of the server startup parameters.		`/dr01/qdt/envs/qaddemo/scripts/base-live-set.pf`
QAD Enterprise Applications code page and collation		utf-8 ICU-UCA
QAD Configurator License Code to register the product and complete the installation		

Installing QAD Configurator on an Oracle database also requires the following information:

Prerequisite Information	Variable	Example
Oracle service network port. Typically, it is 1521.	*`$OraclePort`*	1521
Oracle instance SID where QAD Enterprise Applications is installed.	*`$OracleSID`*	
Oracle DBA username/password combination.	*`$OraDBAUser/`* *`$OraDBAPass`*	admin/qad
Oracle username/password for the QAD Enterprise Applications schema owner. Typically, the username/password combination is qad/qad.	*`$OraMFGUser/`* *`$OraMFGPass`*	qad/qad

Installation Steps

The QAD Configurator installation requires the following general steps:

- Install QAD Configurator Files
- Create QAD Configurator Databases (Progress Database)
- Create Start/Stop Scripts for the Configurator Production Databases
- Modify QAD Configurator AppServer API File
- Configure the Site-Specific Data Creation Feature
- Compile QAD Configurator AppServer API Files (Progress Database)
- Modify the QAD .NET UI .pf File (Progress Database)
- Modify the QAD .NET UI AppServer Configuration

- Modify WebSpeed Configurations
- Set Up QAD Configurator .NET UI Plug-In
- Set Up QAD Configurator Process Maps
- Register QAD Configurator
- Configure WebSpeed Settings
- Modify Desktop Telnet Scripts
- Set QAD Desktop Connection Timeout (Optional)
- Apply the Configurator-TrM Integration Patch

Install QAD Configurator Files

The installation steps assume that you have a single-tier QAD Enterprise Applications environment and are installing all the following Configurator components on one server:

- QAD Configurator database, system data, and toolset files
- QAD Configurator AppServer code
- QAD Configurator WebSpeed code
- QAD Configurator .NET plug-in
- QAD Configurator process maps
- QAD Configurator Browse Collections

If you are installing the components separately on several machines, run the installation script on each machine and enter appropriate values when prompted to install the corresponding components.

1 Launch the installation script located under the `install` directory in the installation media:

```
./install.ksh
```

2 Read the on-screen instructions and use the following table to enter the appropriate values for script execution.

Table 1.2
Install Script Steps

At this step	Do this
Welcome screen	Press Enter; then press the SPACEBAR to read the license agreement or press q to jump to the end of the license agreement.
Do you accept all the terms of the preceding license Agreement?	Enter y.
Where do you want to save the install log file?	Accept the default or enter a new location; for example, `/dr01/qad/instlog`. If the installation log directory you provided does not exist, the script will prompt you to create it. Enter y.

At this step	Do this
Unable to locate `instcpd.ini`. ... Continue anyway?	If the script cannot locate the file `instcpd.ini`, it will ask you whether to proceed. If this is your initial QAD Configurator install, Enter y here to create this file. If this is an existing install, consider answering n here and locating the file. It contains useful previous installation information.
Enter Progress installation directory.	Enter the directory path where Progress is installed; for example, `/dr01/progress/dlc`.
Do you wish to install database, system data, and toolset files for Configurator now?	Enter y. If you do not want to install the components on the current machine, enter n.
• Where do you want to install QAD Configurator database server?	Enter the directory where you want to install QAD Configurator database server; for example, `/dr01/qad/configurator`. This directory is identified as *`CfgDBSvrInstallDir`* in this guide. When creating the directory, the install script will automatically create `cpd/database` subdirectories under it.
• Which edition of QAD Enterprise Applications are you running?	Enter the edition of QAD Enterprise Applications you are running: EE for Enterprise Edition and SE for Standard Edition.
Do you wish to install QAD Configurator AppServer files now?	Enter y. If you do not want to install the component on this machine, enter n.
• Where do you want to copy the QAD Configurator AppServer files?	Enter the directory where you want to install QAD Configurator AppServer API files; for example, `/dr01/qad/configurator`. This directory is identified as *`CfgAppServerDir`* in this guide. When creating the directory, the install script will automatically create a `cpd` subdirectory under it.
Do you wish to install QAD Configurator WebSpeed files now?	Enter y. If you do not want to install the component on this machine, enter n.
• Where do you want to copy the QAD Configurator WebSpeed files?	Enter the directory where you want to install the Configurator WebSpeed files; for example, `/dr01/qad/configurator`. This directory is identified as *`CfgWebSpeedDir`* in this guide. When creating the directory, the install script will automatically create a `cpd` subdirectory under it.
Do you wish to install QAD .NET UI plug-ins for Configurator now?	Enter y. If you do not want to install the component on this machine, enter n.

At this step	Do this
• Enter the directory where QAD .NET UI plug-ins are located.	Enter the directory where you want to install QAD .NET plug-in for Configurator. For QAD .NET UI 2.5.3, you can specify any directory. For other .NET UI versions such as 2.7.x and 2.8.x, enter *TomcatInstallDir*/webapps/qadhome/packages.
• Which QAD .NET UI version are you running?	Specify the version of QAD .NET UI you are running. You can look up this information in the `version.net` file under the Tomcat server installation directory.
Do you wish to install QAD Configurator process maps now?	Enter y. If you do not want to install the component on this machine, enter n.
• Enter the directory where QAD .NET UI process maps are located.	Enter the directory where you want to install Configurator .NET process map files. Typically, it is *TomcatInstallDir*/webapps/*qaduiConfig*.
Do you wish to install QAD Configurator Browse Collections now?	Enter y. If you do not want to install the component on this machine, enter n.
• Enter the directory where QAD .NET UI browse collections are located.	Enter the directory where you want to install Configurator browse collection files. Typically, it is *TomcatInstallDir*/webapps/qadhome/configurations/*qaduiConfig*.
• Which QAD .NET UI version are you running?	Specify the version of QAD .NET UI you are running. You can look up this information in the `version.net` file under the Tomcat server installation directory.
Proceed using these values?	If the values you entered are correct, accept the default. Otherwise, enter n and you will be returned to the point where you specified the Progress directory so you can modify previous values.

3 The install script executes. When execution is complete, press Enter to end the script.

4 At the end of the script, the name and location of the installation log file display. You can open the log file in a text editor to check for errors if necessary.

Create QAD Configurator Databases (Progress Database)

Use these steps if you are installing QAD Configurator on Progress database. For Configurator Oracle implementation, see “Create QAD Configurator Databases (Oracle Database)” on page 14.

Create the QAD Configurator Empty Database

The `cpdempty` database is used to create the production QAD Configurator database.

Use the utility program that comes with QAD Configurator to perform the following steps:

1 Launch MFG/UTIL for Configurator. Execute `cpdutil` from *`CfgAppServerDir`*`/cpd/`.

Note MFG/UTIL assumes that the QAD Enterprise Applications database uses the utf-8 code page. If your QAD Enterprise Applications database uses a different code page, open `cpdutil` in a text editor and modify the startup parameters — -cpinternal, -cpstream, -cpcoll — to use the same code page used by your QAD Enterprise Applications database.

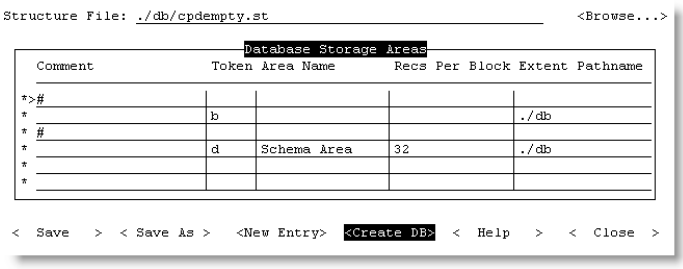
Note A warning message may appear telling you that the code page you use is not supported for TTY clients and that you may corrupt files or databases. You can safely ignore it.

2 The MFG/UTIL screen displays. Select Configure|QAD ERP Guided Setup.

3 The Operation Sets screen displays. Select Create Configurator Progress Empty DB from the Operation Set drop-down list; then choose Run Set.

4 The QAD Database Builder screen displays. Choose Create DB.

Fig. 1.3
QAD Database Builder



5 The Create/Copy Database dialog box displays. Choose other database and specify an empty database that uses the same code page as your QAD Enterprise Applications database to copy.

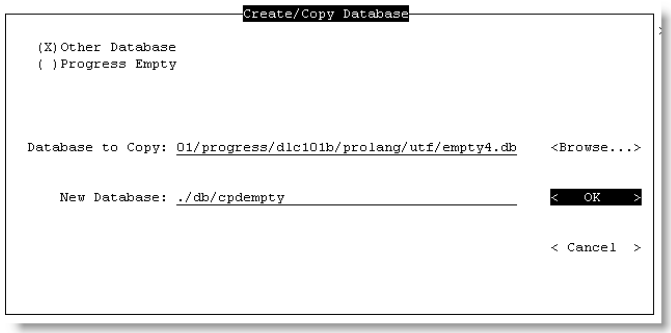
There is an empty database for each locale in the *`ProgressInstallDir`*`/prolang` directory; for example, `/dr01/progress/dlc/prolang/eng/empty4` for the ISO8859-1 code page.

If you want to use the UTF-8 code page, specify *`ProgressInstallDir`*`/prolang/utf/empty4.db` to copy.

For information on creating databases using other code pages, see Progress documentation.

When finished, choose OK.

Fig. 1.4
Create Empty Database



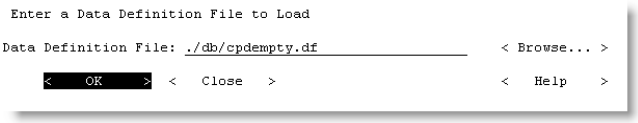
6 The database creation process begins. The QAD Create Database Monitor screen displays progress and error status. When database creation is complete, choose Close to exit.

7 You return to the QAD Database Builder screen; choose Close to exit.

8 The Connect Database screen displays. Connect to `cpdempty.db` under *`CfgDBSvrInstallDir`*`/cpd/db` and choose OK.

9 The Load Data Definitions screen displays. The database schema files (`.df`) loaded contain the table, field, and index definitions for your QAD Configurator database. Choose OK.

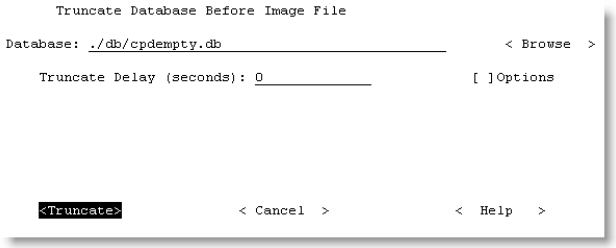
Fig. 1.5
Load the QAD Configurator Data Definitions



10 A load screen displays. When the load is complete, choose Close in the QAD Log screen.

11 The Truncate Database Before Image File screen displays. Choose Truncate.

Fig. 1.6
Truncate BI File



12 The truncate process begins showing progress and error status. When file truncate is complete, choose Close to exit.

13 You return to the Operation Sets screen. Choose Close to exit.

The QAD Configurator empty database is created.

Build the QAD Configurator Production Database

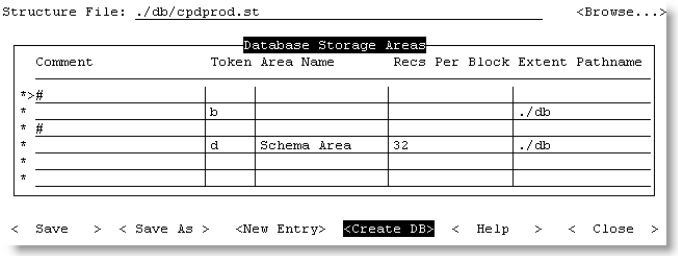
These steps create a production database.

1 Select Configure|QAD ERP Guided Setup.

2 The Operation Sets screen displays. Select Create Configurator Progress Production DB from the Operation Set drop-down list; then choose Run Set.

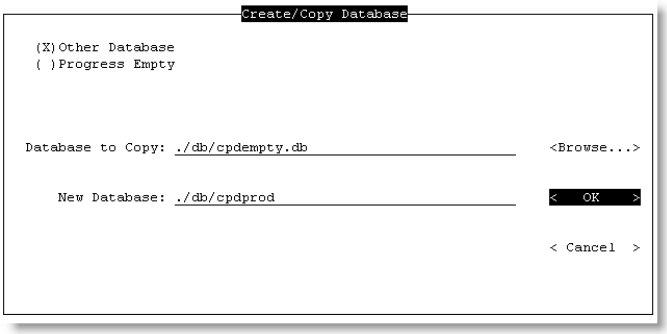
3 The QAD Database Builder screen displays. Choose Create DB.

Fig. 1.7
QAD Database Builder



4 The Create/Copy Database dialog box displays. Choose OK.

Fig. 1.8
Create Production Database



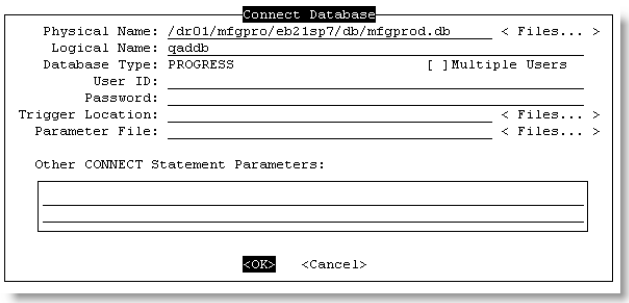
5 The database creation process begins. The QAD Create Database Monitor screen displays showing progress and error status. When database creation is complete, choose Close to exit.

6 You return to the QAD Database Builder screen; choose Close to exit.

7 The Connect Database screen displays. Connect to *QADERPInstallDir*/db/mfgprod.db and choose OK.

Note Close all other applications connected to mfgprod.db or select the Multiple Users mode.

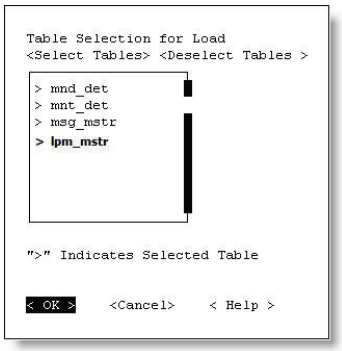
Fig. 1.9
Connect to the QAD Enterprise Applications Production Database



8 The QAD Log screen displays. Choose Close.

9 The Table Selection for Load screen displays. With all the tables in the list selected, choose OK.

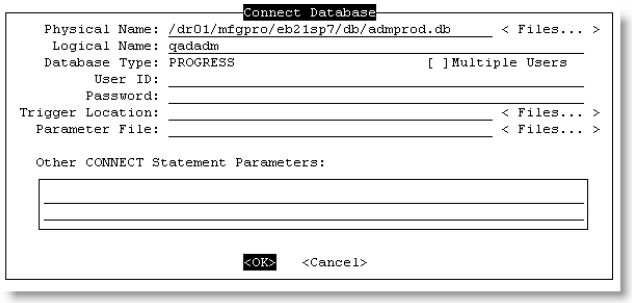
Fig. 1.10
Table Selection for Load



10 The QAD Log screen displays the records of the selected tables that are being processed. When the data load is complete, choose Close to exit.

11 The Connect Database screen displays. Connect to *QADERPInstallDir*/db/admprod.db and choose OK.

Fig. 1.11
Connect to the QAD Enterprise Applications Administration Database



12 The QAD Log screen displays. Choose Close.

13 The Table Selection for Load screen displays. With all the tables in the list selected, choose OK.

14 The QAD Log screen displays. Choose Close.

15 The Connect Database screen displays. Connect to *CfgDBSvrInstallDir*/cpd/db/cpdprod.db and choose OK.

16 The QAD Log screen displays. Choose Close.

17 The Table Selection for Load screen displays. With all the tables in the list selected, choose OK.

18 The QAD Log screen displays. Choose Close.

19 The Truncate BI File screen displays. Choose Truncate.

20 The truncate process begins. The QAD Database Monitor screen displays progress and error status. When file truncate is complete, choose Close to exit.

21 You return to the Operation Sets screen. Choose Close to exit.

22 If you are not installing the English language version, manually load the .d files under *./LanguageCode/*mfg and *./LanguageCode/*adm into the mfgprod.db and admprod.db databases respectively, where *LanguageCode* is the implemented language code. Supported language codes are listed as follows:

Language Code	Language
ch	Simplified Chinese
tw	Traditional Chinese
cs	Castilian Spanish
ls	Latin American Spanish
du	Dutch
fr	French
ge	German
it	Italian
jp	Japanese
pl	Polish
po	Portugese

The QAD Configurator production database is created.

Create QAD Configurator Databases (Oracle Database)

Use these steps if you are installing QAD Configurator on Oracle database. For Configurator Progress implementation, see “Create QAD Configurator Databases (Progress Database)” on page 9.

Create Oracle schema

1 Modify *CfgDBSvrInstallDir*/cpd/db/tablespace.sql based on your system installation environment.

```
/*****************************************************************/
CREATE TABLESPACE configurator
DATAFILE '{ORACLE_BASE}/oradata/{DB_NAME}/configurator.dbf' SIZE 25M
AUTOEXTEND ON NEXT 25M MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
ONLINE;
ALTER USER qad QUOTA UNLIMITED ON configurator;
/*****************************************************************/
```

Replace the file path in bold with the absolute path of the new Oracle DataFile.

Replace the username in bold with the Oracle username *$OraMFGUser*. If this username has been modified, replace all occurrences of default FOREIGN-OWNER “QAD” in file *CfgDBSvrInstallDir*/cpd/db/oraprocon.df with the current username.

2 Run Oracle SQLPlus and connect to the QAD Enterprise Applications schema instance as DBA. Run script *CfgDBSvrInstallDir*/cpd/db/tablespace.sql.

```
[oracle@vmlinux ~]$ sqlplus $OraDBAUser/$OraDBAPass@localhost:$OraclePort/$OracleSID as
```

```
sysdba
SQL> @CfgDBSvrInstallDir/cpd/db/tablespace.sql
```

3 Run Oracle SQLPlus and connect to the instance as QAD Enterprise Applications schema owner. Run script *CfgDBSvrInstallDir*/cpd/db/procon.sql to create Configurator schema.

```
[oracle@vmlinux ~]$ sqlplus $OraMFGUser/$OraMFGPass@localhost:$OraclePort/$OracleSID
SQL> @CfgDBSvrInstallDir/cpd/db/procon.sql
```

Create Progress Schema Holder

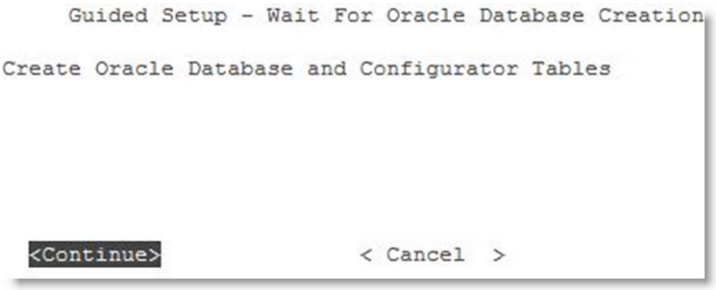
1 Run CPDUtil from *CfgDBSvrInstallDir*/cpd.

2 Create Configurator Oracle empty database.

a In the CPDUtil main screen, choose Configure|QAD ERP Guided Setup|Create Configurator Oracle Empty DB.

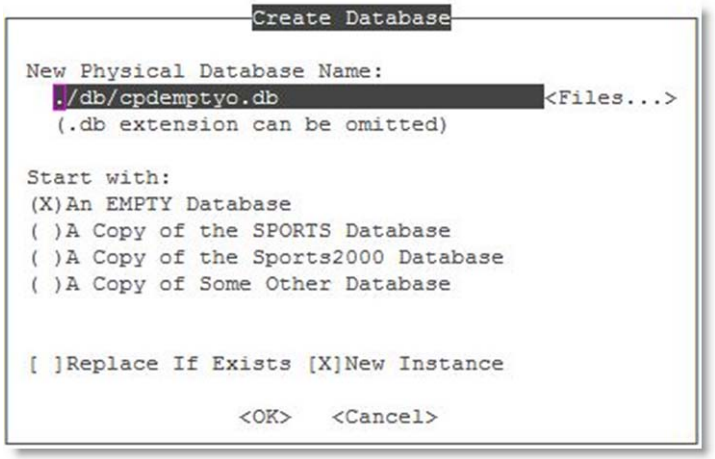
b Wait until Oracle database creation is complete. If no error occurs in the process, choose Continue.

Fig. 1.12
Oracle Database Creation



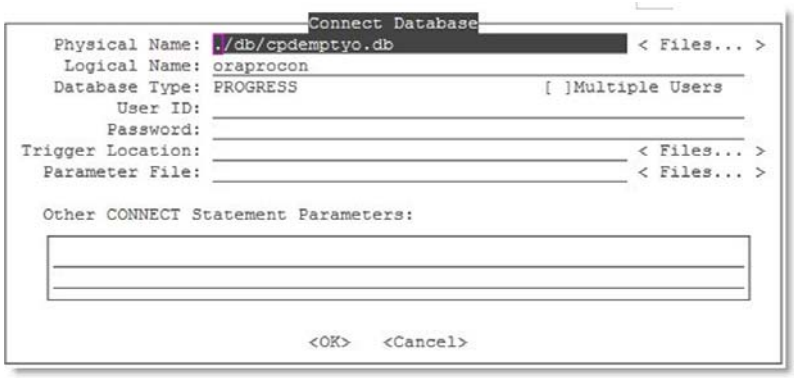
c Create Configurator Oracle schema holder empty database. Use the default or specify a new physical database path and file name, referred to as $EMPTY_ORASH_DB in subsequent steps.

Fig. 1.13
Create Database



d Connect the empty database (`$EMPTY_ORASH_DB`) in the single-user mode. Accept all the defaults if you did not change the database path and file name in the previous step.

Fig. 1.14
Connect Empty Database

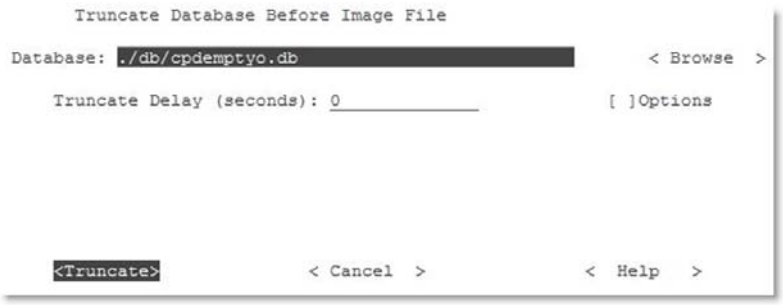


e Load schema file into the empty database. Enter *`CfgDBSvrInstallDir`*`/cpd/db/oraprocon.df` in the Data Definition File field.

Fig. 1.15
Load Data Definition File

f Truncate BI for the empty database.

Fig. 1.16
Truncate BI



3 Create Configurator Oracle production database from the empty database.

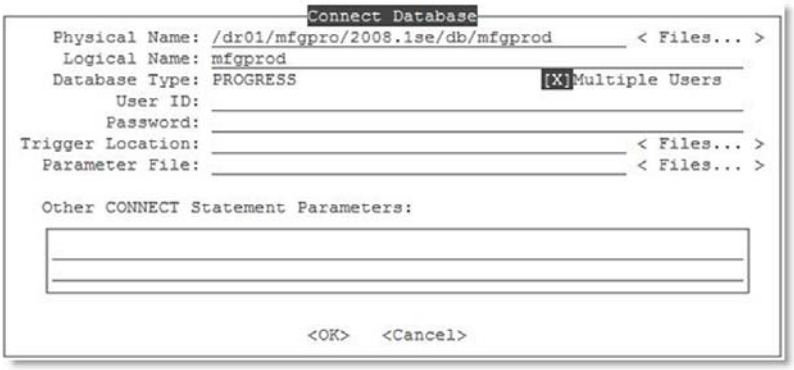
a In CPDUtil main screen, choose Configure|QAD ERP Guided Setup|Create Configurator Oracle Prod DB.

b Enter *`$EMPTY_ORASH_DB`* in the Database to Copy field, and specify the database path and file name for the new production database in the New Database field, referenced as *`$PROD_ORASH_DB`* in subsequent steps.

Fig. 1.17
Create Configurator Oracle Production Database

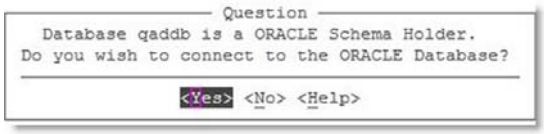
c Connect to the QADDB database to load system data. Make sure that the logical name is mfgprod; otherwise, errors may occur.

Fig. 1.18
Connect to QADDB



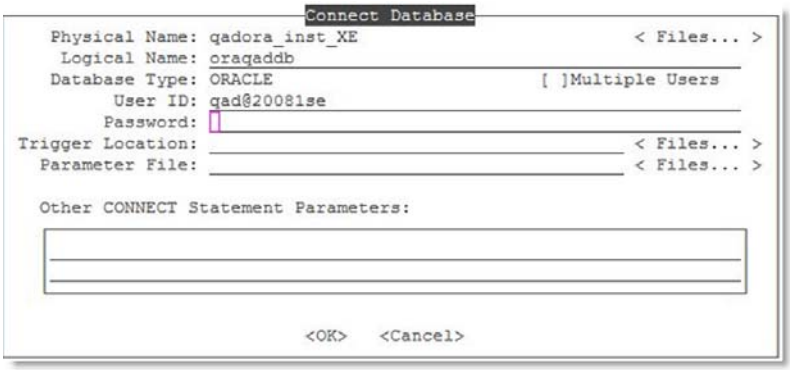
d If the database is an instance of Oracle DataServer, you must choose Yes in the Oracle connection confirmation window.

Fig. 1.19
Oracle Database Connection Confirmation



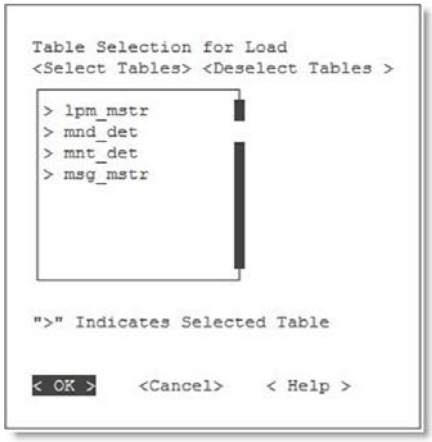
e Connect to the Oracle database instance. Enter *$OraMFGUser@$OracleSID* as user ID and *$OraMFGPass* as Password.

Fig. 1.20
Connect Oracle Database



f In the Table Selection screen, select all tables and choose OK.

Fig. 1.21
Select Tables



g Repeat steps c through f for the QADADM to load admin data.

h Repeat steps c through f for the Configurator production database. Enter *$PROD_ORASH_DB* in the Physical Name field and use mfgprod as the logical name when connecting the database. Select all tables for load in the Table Selection screen.

Create Start/Stop Scripts for the Configurator Production Databases

1 For multi-tier deployment of Configurator, register the Configurator production database server as a service in the `/etc/services` file. For example, add the following line to the file:

```
cfgproddb 30024/tcp #Configurator production DB
```

Note Skip this step if you are installing all the Configurator components on a single machine.

2 Create the Configurator database start/stop script files. Two sample files, `start.cpd` and `stop.cpd`, were copied from the `cpd_db` directory in the installation media to the *`CfgDBSvrInstallDir`*`/cpd` directory by the install scripts. Replace key variables in these files with correct values to reflect your environment parameter values on your server. Key variables to replace are highlighted in bold as follows:

- start.cpd

```
#!/bin/sh
# Script to start database servers.
# tokens:
# &DLC = Progress Directory
# &LOOP-DB-START = start of database loop
# &LOOP-DB-END = end of database loop
# &START-SERVER = command line to start current DB in database loop
DLC=ProgressInstallDir;export DLC
PATH=$PATH:$DLC;export PATH
PROMSGS=$DLC/promsgs;export PROMSGS
PROTERMCAP=$DLC/protermcap;export PROTERMCAP
$DLC/bin/_mprosrv CfgDBSvrInstallDir/cpd/db/cpdprod -L 8000 -c 350 -B 1000 -S
CfgProdDBServiceName -N TCP -cpinternal InternalCodePage -cpstream StreamCodePage -
cpcoll CollationTable
```

Note If you are installing all the Configurator components on a single machine, exclude the following parameters from the scripts:

```
-S CfgProdDBServiceName -N TCP
```

- Stop

```
#!/bin/sh
# Script to stop database servers.
# tokens:
# &DLC = Progress Directory
# &LOOP-DB-START = start of database loop
# &LOOP-DB-END = end of database loop
# &STOP-SERVER = command line to shutdown current DB in database loop
DLC=ProgressInstallDir;export DLC
PATH=$PATH:$DLC;export PATH
PROMSGS=$DLC/promsgs;export PROMSGS
PROTERMCAP=$DLC/protermcap;export PROTERMCAP
$DLC/bin/_mprshut CfgDBSvrInstallDir/cpd/db/cpdprod -by
```

ProgressInstallDir. The Progress installation directory.

CfgDBSvrInstallDir. The Configurator database server installation directory, not including the cpd directory automatically appended by the install script.

CfgProdDBServiceName. The Configurator production database service name you registered in the `/etc/services` file in step 1.

InternalCodePage. The code page used in the memory; for example, ISO8859-1.

StreamCodePage. The code page used for stream I/O; for example, utf-8.

CollationTable. The collation used with the code page in the memory; for example, ICU-UCA.

For information on code pages and collation tables, see Progress documentation on *OpenEdge Development: Internationalizing Applications*.

Modify QAD Configurator AppServer API File

From the directory where you installed Configurator .NET UI API files as described in “Install QAD Configurator Files” on page 7, open the `pcparm.i` file under the `cpd/cop_xrc` subdirectory using a text editor. Specify mfgpro and mfgversion values according to your QAD Enterprise Applications version as listed below:

QAD Enterprise Applications Version	mfgpro Value	mfgversion Value
eB2.1 SP4	eb2.1	4
QAD 2007	eb2.1	5
QAD 2007.1	eb2.1	6
QAD 2008 Standard	eb2.1	7
QAD 2008.1 Standard	eb2.1	8
QAD 2008.1 EE	eb3	3
QAD 2009 EE	eb3	4
QAD 2009 SE	eb2.1	9
QAD 2009.1 EE	eb3	5
QAD 2010 EE	eb3	6
QAD 2010.1 EE	eb3	7

Configure the Site-Specific Data Creation Feature

The site-specific data creation feature lets you specify, for each configurable item, whether to create item-site data along with item master records when the system generates variants from the configurable item, and if so, which sites to use to create item-site data.

By default, this feature is disabled. You can switch it on by editing the `pcparm.i` file from `cpd/cop_xrc` under the directory where you installed Configurator .NET UI API files and setting the site-enabled variable to Yes.

Compile QAD Configurator AppServer API Files (Progress Database)

Use these steps if you are installing QAD Configurator on Progress database. For Configurator Oracle implementation, see “Compile QAD Configurator AppServer API Files (Oracle Database)” on page 22.

1. Launch MFG/UTIL for Configurator. Execute `cpdutil` from *`CfgAppServerDir`*`/cpd/`.

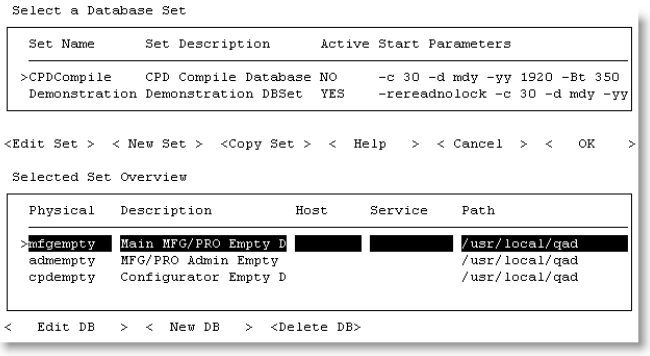
   Note MFG/UTIL assumes that the QAD Enterprise Applications database uses the utf-8 code page. If your QAD Enterprise Applications database uses a different code page, open `cpdutil` in a text editor and modify the startup parameters — -cpinternal, -cpstream, -cpcoll — to use the same code page used by your QAD Enterprise Applications database.

   A warning message may appear telling you that the code page you use is not supported for TTY clients and that you may corrupt files or databases. You can safely ignore it.

2. In MFG/UTIL, select Configure|Database Set Maintenance.

3. The Database Set Configuration screen displays. In the database set list, select CPDCompile.

Fig. 1.22
Database Set Configuration

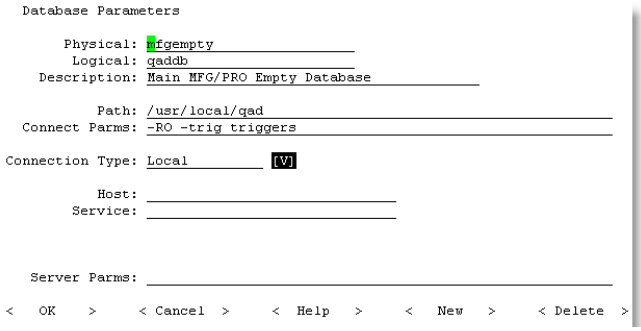


4 Perform the following steps to configure the client view of the cpdempty, mfgempty, and admempty databases:

a Select a database and choose Edit DB.

b When the Client Database Parameters screen displays, complete the fields using the field descriptions as a guide.

Fig. 1.23
Edit Database Parameters



Physical. Physical name of the database; accept the default if there is one.

Path. If you choose a Local connection type, enter or accept the full path to the directory containing the database file (extension `.db`). Specify the correct path for the three databases respectively with mfgempty being the first:

Physical	Description	Path
mfgempty	QAD Enterprise Applications empty DB	*`QADERPInstallDir`*`/db`
admempty	QAD Enterprise Applications admin DB	*`QADERPInstallDir`*`/db`
cpdempty	Configurator Empty DB	*`CfgDBSvrInstallDir`*`/cpd/db`

Accept the defaults for the rest of the fields.

Connection Type. If the Configurator database is installed on the same machine, choose Local; otherwise, choose ClientServer.

Host. If you choose a client/server connection, enter the host name of the machine where the database is located.

Service. For client/server connections, enter a service name for the database that corresponds to an entry in your `\etc\services` file. You must make the entries in the services file separately; the installation does not change this file.

c When ready, choose OK to continue.

5 Once you have configured each database in the set, verify the database set in the Selected Set Overview frame. Click OK in the Database Set Configuration screen.

6 In MFG/UTIL, select Programs|Compile Procedures.

7 The QAD Compiler screen displays. Edit the compile PROPATH as follows:

`CfgAppServerDir/cpd/cop_xrc: QADERPInstallDir/qxtend/xrc: QADERPInstallDir/xrc`

`CfgAppServerDir` is where you installed Configurator AppServer files.

8 If you are not installing the English language version, change the language code to your installation language.

9 Make sure the compile destination directory is different from the QAD Enterprise Applications installation directory to avoid overwriting standard QAD Enterprise Applications files.

10 Choose Compile. When you are prompted to overwrite the current directory, choose OK to proceed.

11 The QAD Compiler Summary Status screen displays. Choose Continue.

12 The QAD Log screen displays compile progress and error status. When the compile is complete, choose Close to exit.

13 If you are using eB2.1 SP4, QAD 2007, or QAD 2007.1, recompile the QAD Enterprise Applications code and Desktop code with the following in the front of the PROPATH:

`QADERPInstallDir/qxtend/xrc:QADERPInstallDir/qra/xrc`

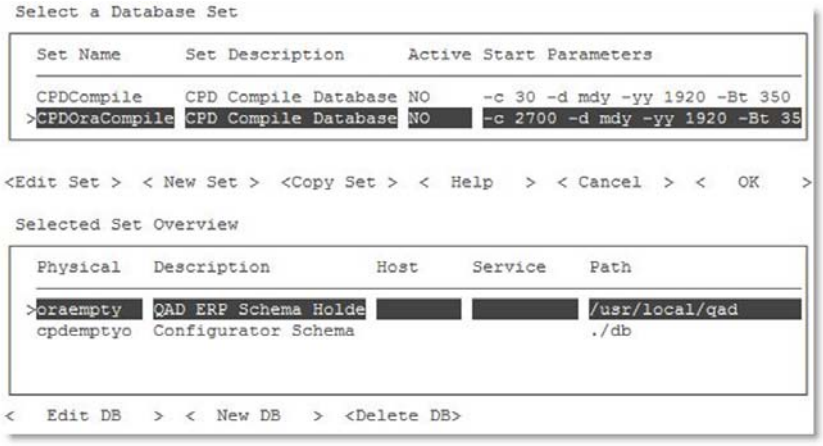
Compile QAD Configurator AppServer API Files (Oracle Database)

If you have an Oracle database, use the following steps instead of those in “Compile QAD Configurator AppServer API Files (Progress Database)” on page 20.

1 Run CPDUtil from *`CfgAppServerDir`*`/cpd` and choose Configure|Database Set Maintenance.

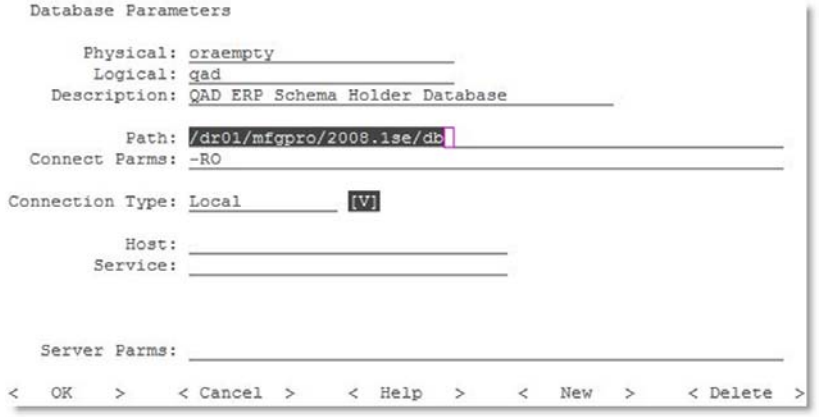
2 Select the CPDOraCompile database set and then the oraempty database; then choose Edit DB.

Fig. 1.24
Configure Database Set



3 In the Database Parameters screen, modify the path to *QADERPInstallDir*/db; then choose OK.

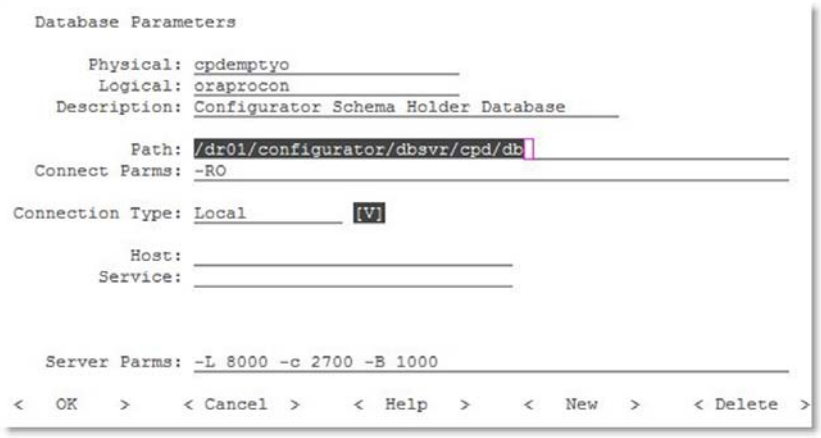
Fig. 1.25
Select Database Set



4 Select the oraempty database from the CPDOraCompile database set and choose Edit DB.

5 In the Database Parameters screen, modify the path to the Configurator schema holder empty database path, *CfgDBSvrInstallDir*/cpd/db; then choose OK.

Fig. 1.26
Select Database Set

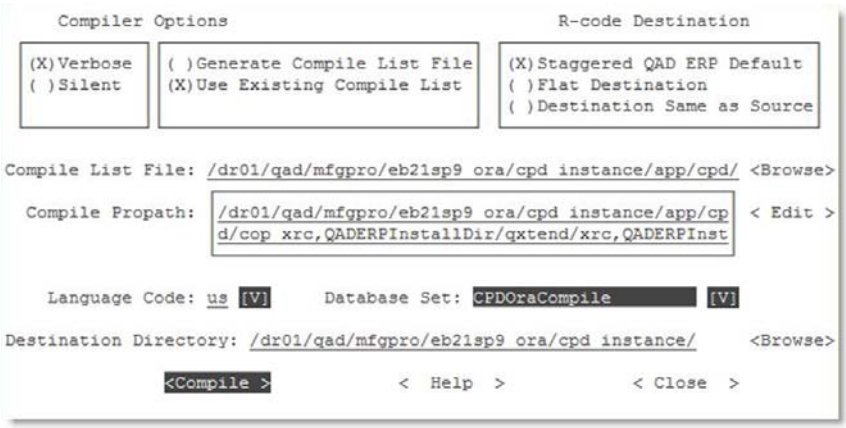


6 Back in the CPDUtil main screen, choose Programs|Compile Procedure.

7 In the Compile Options screen, specify CPDOraCompile as the database set and modify the compile path to the following:

CfgAppServerDir/cpd/cop_xrc,*QADERPInstallDir*/qxtend/xrc,*QADERPInstallDir*/xrc,.

Fig. 1.27
Compile Options



Modify the QAD .NET UI .pf File (Progress Database)

If you are installing QAD Configurator on Oracle database, see “Modify the QAD .NET UI .pf File (Oracle Database)” on page 25.

Modify the existing QAD .NET UI .pf `base-live-set.pf` file to include the QAD Configurator production database. For the location of the .NET UI .pf file, refer to the Configurator installation worksheet prepared during “Configurator Installation Preparations” on page 5.

```
-db QADERPInstallDir/db/mfgprod -ld qaddb -H DBSvrHostName -S
mfgprodDBServiceName -trig triggers

-db QADERPInstallDir/db/admprod -ld qadadm -H DBSvrHostName -S
admprodDBServiceName -trig triggers

-db CfgDBSvrInstallDir/cpd/db/cpdprod -ld procon -H DBSvrHostName -S
CfgProdDBServiceName -trig triggers
```

QADERPInstallDir. QAD Enterprise Applications database server installation directory.

DBSvrHostName. Database server host name. Specify this parameter only if you are using the Client-Server database connection type.

mfgprodDBServiceName, admprodDBServiceName. Service name of the QAD Enterprise Applications production database and administration database respectively, registered in the `/etc/services` file. Specify these parameters only if you are using the Client-Server database connection type.

CfgProdDBServiceName. Service name of the Configuration production database, registered in the `/etc/services` file. Specify these parameters only if you are using the Client-Server database connection type.

Note Specify `-H DBSvrHostName -S mfgprodDBServiceName` only when your database connection type is Client-Server.

Note If you are using the utf-8 code page, make sure the following is added to the .pf file:

```
-cpinternal utf-8  -cpstream utf-8 -cprcodeout utf-8  -cpcoll ICU-UCA
```

Modify the QAD .NET UI .pf File (Oracle Database)

Use these steps if you are installing QAD Configurator on Oracle database. For Configurator Progress implementation, see “Modify the QAD .NET UI .pf File (Progress Database)” on page 24.

Edit the QAD .NET UI .pf file and add the following line to it:

```
-db CfgDBSvrInstallDir/cpd/db/cpdprodo -ld oraprocon -H DBSvrHostName -S
CfgProdDBServiceName -trig triggers
-db procon -dt ORACLE -U $OraMFGUser@$OracleSID -P $OraMFGPass -c 2700
```

Modify the QAD .NET UI AppServer Configuration

1. Open the `ubroker.properties` file under the *`ProgressInstallDir/`*`properties` directory using a text editor and locate the [UBroker.AS.*`qaduiASService`*] section.
2. Append the following QAD Configurator directories to the PROPATH.

```
CfgAppServerDir/cpd:
CfgAppServerDir/cpd/LanguageCode:
CfgAppServerDir/cpd/cop_xrc:
QADERPInstallDir/qxtend
```

3. Add the following directories before the QAD Enterprise Applications path:
   - For 32 bit platforms:

QADERPInstallDir/qra/qra.pl:

- For 64 bit platforms:

QADERPInstallDir/qra/qra64.pl:

4 Set the following:

```
srvrShutdownProc=mfaishut.p
srvrStartupProc=mfaistrt.p
```

Modify WebSpeed Configurations

1 Add the WebSpeed service configuration for Configurator.

Note Configurator's WebSpeed files only support Apache servers.

a Open the `ubroker.properties` file under the *`ProgressInstallDir/`*`properties` directory using a text editor.

b Create a new WebSpeed service configuration section in the file for Configurator. Use the following sample as reference.

```
[UBroker.WS.CfgWSService]
  srvrLogFile=/qad/web/server/logs/cfa92b/testcfa92bui1.server.log
  brokerLogFile=/qad/web/server/logs/cfa92b/testcfa92bui1.broker.log
  portNumber=48396
  initialSrvrInstance=1
  maxSrvrInstance=15
  autoTrimTimeout=600
  appserviceNameList=testcfa92bui1
  controllingNameServer=NS1
  environment=testcfa92bui1
  uuid=561db2f13e5a0142:12dacd1:11739307fc5:-8000
  description=WebSpeed Transaction server for testcfa92bui1
srvrStartupParam =-pf /qad/web/workingdirectories/testcfa92bui1.pf -p
CfgWebSpeedDir/cpd/src/web/objects/web-disp.p -weblogerror -rereadnolock -cpstream
utf-8
  PROPATH=
CfgWebSpeedDir/cpd/:CfgWebSpeedDir/cpd/src:CfgWebSpeedDir/cpd/proxy:QADERPInstallD
ir:QADERPInstallDir/xrc:QADERPInstallDir/qra/qra.pl:QADERPInstallDir/qra/xrc:.
```

Key parameters and values are highlighted in bold:

CfgWSService. Specify a QAD .NET UI WebSpeed service for Configurator.

uuid. Generate a uuid using the genuuid Progress command.

CfgWebSpeedDir. The directory where you installed Configurator WebSpeed files during install scripts execution.

Note Make sure the following directory is specified at the end of the PROPATH.

QADERPInstallDir/qra/qra.pl:*QADERPInstallDir*/qra/xrc

Note In case there is no `qra/xrc` in your environment, copy `/qra/qra.zip` from the installation media and extract it to the *`QADERPInstallDir/`*`qra` directory.

Note If you are using a 64-bit platform, specify the following at the end of the PROPATH:

QADERPInstallDir/qra/qra64.pl:*QADERPInstallDir*/qra/xrc

Note The `.pf` file here should have the same database connection as the AppServer `.pf` file.

2 Modify Apache Web server settings. Edit the Apache configuration file `httpd.conf` file to create an alias to point to *`CfgWebSpeedDir/`*`cpd/htdocs`:

```
Alias /CfgAlias/ "CfgWebSpeedDir/cpd/htdocs/"
```

```
<Directory "CfgWebSpeedDir/cpd/htdocs">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Note To make sure the Web page uses the utf-8 code page, set AddDefaultCharset to off or utf-8 .

3. Make sure `wspd_cgi.sh` is installed in `cgi-bin`. If the file is not there, copy it from *`ProgressInstallDir`*`/cgi-bin`.
4. Restart the Web server, AdminServer, and WebSpeed Workshop. Here is an example of the WebSpeed Workshop URL:

   `http://`*`HostServerName`*`/cgi-bin/wspd_cgi.sh/WService=`*`CfgWSService`*`/workshop`
5. Compile Configurator WebSpeed files. On the left of the WebSpeed WorkShop UI, click PROPATH.
6. Enter `compile.html` in the Find File text box and click Submit Query.
7. Click the first matching file that displays.
8. On the next page, select `compile.html` and click the Compile icon.
9. Click the Run icon; then select all the files and click the Compile icon.

Set Up QAD Configurator .NET UI Plug-In

1. If you are running in QAD .NET 2.7.1 through 2.9.2, modify *`TomcatInstallDir`*`/webapps/qadhome/packages/ plugins/manifest.qpkg` to add the following before `</package>`.

   ```
   <package path="" ref="${Repos}/plugins/QAD.Configurator/manifest.qpkg" />
   ```
2. Skip this step if you are running in QAD .NET UI 2.5.3. For other versions, `QAD.Configurator` will be automatically installed after you restart the QAD .NET client.

   For QAD Enterprise Edition, run System Synchronize and select Resources; then restart the .NET UI after synchronization is complete.
3. If you are running in QAD .NET UI 2.5.3, manually unzip the `QAD.Configurator/data.zip` file to the default client's directory:

   ```
   C:\Program Files\QAD\QAD Enterprise Applications 2007.1
   ```

Set Up QAD Configurator Process Maps

If you are running QAD 2008.1 EE and later Enterprise Editions, perform the following steps:

1. Run *`CfgAppServerDir`*`/cpd/cpdutil` to launch MFG/UTIL, where *`CfgAppServerDir`* is the directory under which you installed Configurator AppServer files.
2. Choose Upgrade Configurator|Merge ProcessMap XML. The CFG Process Map screen displays.
3. Enter the following values:

Target Directory. Enter the QAD Configurator process map installation directory, which is normally the Desktop .NET UI installation directory:

TomcatInstallDir/webapps/qaduiConfig.

Source Directory. Enter *CfgAppServerDir*/cpd.

Backup Directory. Enter a directory where to back up the original process maps. It can be any directory.

If you are using QAD .NET UI 2.5.3, perform the following steps:

1. Go to the Desktop menu properties directory. By default, it is *TomcatInstallDir*/webapps/qadhome/configurations/default/menus.
2. Edit plugin-menu.xml. Insert the following child node into the <ShellMenu> node:

```
<!-- Configurator Process Map Menu -->
<ShellMenuItem key="process.Configurator" label="Configurator" image="ProcessMap">
   <Command type="QAD.Commands.WebBrowserCommand">
      <Parameter value="${DesktopBaseUrl}/ProcessViewer.jsp?BreadCrumb=
yes&amp;apisource=AppShell&amp;ProcessName=Config_Configurator" type=
"System.Uri,System"/>
      <Property name="Title" value="Configurator"/>
   </Command>
</ShellMenuItem>
```

3. Restart the Tomcat server. QAD Configurator process maps are installed in the .NET UI.

Register QAD Configurator

Use License Registration Menu (36.16.10.1) in QAD Enterprise Applications to add the license code for QAD Configurator.

Configure WebSpeed Settings

1. Launch QAD Configurator and configure the WebSpeed settings in Configurator Control:

   WebSpeed URL. Enter the WebSpeed Workshop URL in the following format:

   http://*WebSpeedServerHostName*/cgi-bin/wspd_cgi.ksh/WService=*WebSpeedBrokerName*

   Static Web Context URL. Enter the URL address in the format of http://*WebServerHostName*/*CfgAlias* where the scripts, images, and styles folders are published, where *CfgAlias* is the Apache server Alias configuration that points to *CfgWebSpeedDir*/cpd/htdocs

   Web Connection Timeout. Specify the amount of time allowed to try to connect to the Web server before the system stops trying.

   Web Configuration Path. Specify the path where to store the AppServerConnection.xml file. You must first create the directory for storing the configuration file under *CfgWebSpeedDir* and give it read and write permissions.

2. Restart the WebSpeed server.

Modify Desktop Telnet Scripts

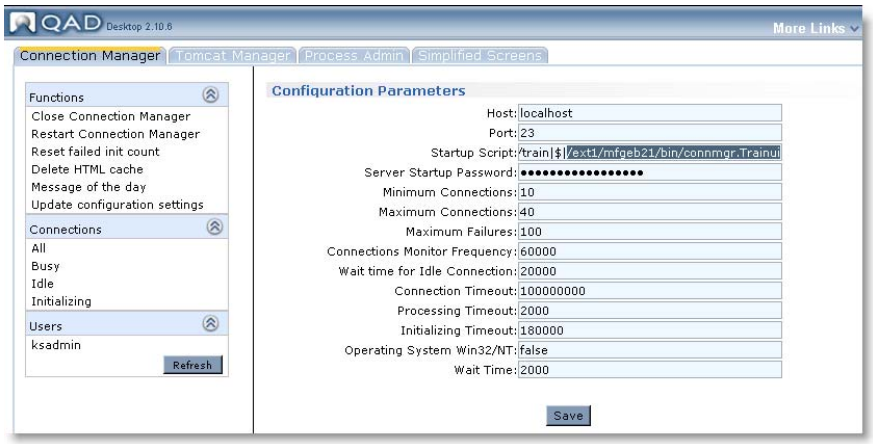
Use the following steps to modify Desktop Telnet scripts to be used by sales order/sales quote line triggers.

1 Log in to Desktop.

2 In Connection Manager, click Update Configuration Settings.

3 In the Configuration Parameters pane, click in the Startup Script field and then move to the end of the entry to find the name of the Telnet script file.

Fig. 1.28
QAD Desktop Configuration Parameters



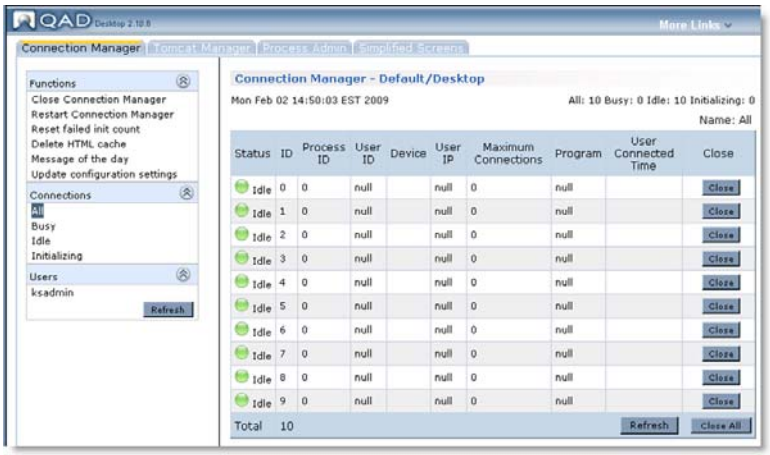
4 Open the Telnet script file in a text editor and make the following changes:

- Add *`AppServerInstallDir`*`/cpd` and *`AppServerInstallDir`*`/cpd/`*`LanguageCode`* to the beginning of the PROPATH, where *`AppServerInstallDir`* is the Configurator AppServer installation directory and *`LanguageCode`* is the implemented language code; for example, `ge`.
- Change `mfwb01a.p` to `mfwb01a_c.p` at the end of the file.

5 Back in Desktop Connection Manager, click Restart Connection Manager to restart it. When a confirmation message appears, click OK to restart all connections.

6 In the Connection pane, click All to display all the connections. Check if all the connections are successfully restarted with status Idle.

Fig. 1.29
All Connections Successfully Started with Idle Status

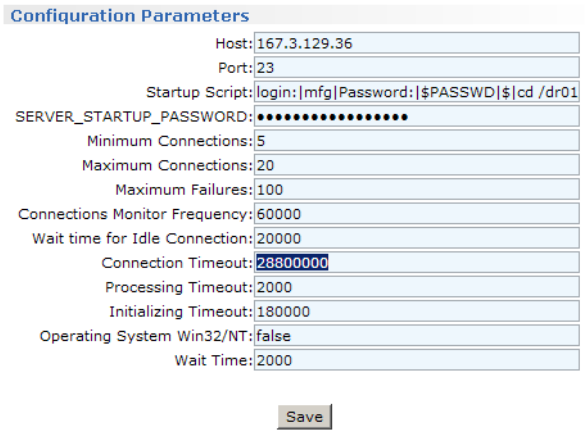


Set QAD Desktop Connection Timeout (Optional)

If a Configurator questionnaire may take a long time to complete, you need to adjust the QAD Desktop connection timeout through QAD Desktop Connection Manager.

To set QAD Desktop connection timeout, access QAD Desktop and go to Update Configuration Settings in Connection Manager. In the Connection Timeout field, set the value to a number (in milliseconds) large enough to allow for sufficient time to complete a Configurator questionnaire.

Fig. 1.30
QAD Desktop Configuration Parameters



Integrate with Trade Management (Optional)

QAD Configurator supports Trade Management (TrM) versions 2.7.6 and 2.8.2. If you want to integrate the Configurator variant item creation process with the QAD Trade Management product, you must apply the corresponding Configurator-TrM integration patch and perform some additional steps. There are two patches under the *`CfgAppServerDir`*`/cpd/trmitemapi` directory for integration with TrM 2.7.6 and 2.8.2 respectively:

- 2.7.6: This patch has been certified against this configuration: QAD 2008.1 SE + QXtend 1.6.2 + TrM 2.7.6.

- 2.8.2: This patch has been certified against this configuration: QAD 2009 SE + QXtend 1.6.2 + TrM 2.8.2.

The following naming conventions are used throughout the subsequent instructions for the Configurator-Trade Management integration.

PatchDir. *CfgAppServerDir*`/cpd/trmitemapi/`*TrMVer*, the directory where the patch was automatically extracted to during the installation process, where *TrMVer* is the TrM version number: 2.7.6 or 2.8.2.

QADERPInstallDir. The directory where QAD Enterprise Applications is installed.

TrMInstallDir. The directory where QAD Trade Management is installed.

QXtendAdapterInstallDir. The directory where QXtend Adapter is installed. Typically, it is *QADERPInstallDir*`/qxtend`.

CfgAppServerDir. The directory where Configurator AppServer API files are located.

Use the following steps to integrate Configurator with Trade Management.

Apply the Configurator-TrM Integration Patch

1. Compile the patch files.
   a. Launch MFG/UTIL for QAD Enterprise Applications from *QADERPInstallDir*.
   b. Choose Procedure|Compile.
   c. In the Compile screen, enter the following:

   R-code Destination. Staggered MFG/PRO Default

   Compile List File. *PatchDir*`/xrc/utcompil.wrk`

   Compile PROPATH .
   PatchDir`/xrc:`*QXtendAdapterInstalldir*`/xrc:`*TrMInstallDir*`/xrc:`*TrMInstallDir*`/us/xrc:`*QADERPInstallDir*`/xrc`

   Destination Directory. *PatchDir*

   d. Execute the compile process.
2. Modify these startup script files:
   - QAD Enterprise Applications client startup script
   - QAD .NET UI Connection Manager startup script
   - QAD .NET UI Telnet startup script

   Add the following to the beginning of the PROPATH in each of these files:

   ```
   PatchDir:PatchDir/us
   ```
3. Import required data.
   a. Run MFG/UTIL for QAD Enterprise Applications from *QADERPInstallDir*.
   b. Choose File|Progress Editor.

c In Progress Editor, type the following command and press F1 to execute it:

d run *PatchDir*/data/applypatch.p.

e Connect to your TrM production database.

f Enter *PatchDir*/data in Input Directory.

g Select tables scontprm and sprgfldi.

h Connect to your TrM production database AGAIN.

4 Validate the patch installation.

a Start QAD Enterprise Applications using the modified client startup script.

b Go to TrM Management Control (7.20.19.1).

c Navigate to page 7 of the control program. You should see the new Run Bulk Upd? field set to Yes by default.

Install QXtend Patch

1 Make sure QXtend 1.6.2 Adapter is installed.

2 Compile QXtend patch files.

a Run MFG/UTIL for QAD Enterprise Applications from *QADERPInstallDir*.

b Choose Procedure|Compile.

c In the Compile screen, enter the following:

R-code Destination. Flat Destination

Important You must choose Flat Destination; otherwise, the Framework files cannot be compiled successfully.

Compile List File. *PatchDir*/xrc/framework.wrk

Compile PROPATH .

PatchDir/xrc:*QXtendAdapterInstalldir*/xrc:*TrMInstallDir*/xrc:*TrMInstallDir*/us/xrc:*QADERPInstallDir*/xrc

Destination Directory. *PatchDir*

d Execute the compile process.

Recompile Configurator AppServer

1 Make sure Configurator AppServer is installed and configured correctly.

2 Turn on Configurator TrM Integration. Edit file *CfgAppServerDir*/cop_xrc/pcparm.i and change this line:

```
&GLOBAL-DEFINE trm-enabled  no
```

to this:

```
&GLOBAL-DEFINE trm-enabled  yes
```

3 Compile Configurator files.

a Run `cpdutil` from *`CfgAppServerDir`*`/cpd/`.

b Choose Procedure|Compile.

c In the Compile screen, enter the following:

R-code Destination. Staggered MFG/PRO Default

Compile List File. $*`CfgAppServerDir`*`/cop_xrc/utcompil.wrk`

Compile Propath .

`CfgAppServerDir``/cop_xrc:`*`PatchDir`*`/xrc:`*`TrMInstallDir`*`/xrc:`*`TrMInstallDir`*`/us/xrc:`*`QXtendAdapterInstalldir`*`/xrc:`*`QADERPInstallDir`*`/xrc`

Destination Directory. *`CfgAppServerDir`*

d Execute the compile process.

4 Modify Configurator AppServer UBroker settings. Add the following to the beginning of the PROPATH:

`PatchDir``:`*`PatchDir`*`/us:`*`TrMInstallDir`*`:`*`TrMInstallDir`*`/us`

5 Restart Configurator AppServer.

Chapter 2

Upgrading QAD Configurator

Overview ***36***

Prior to Upgrading QAD Configurator ***36***

Upgrading Steps ***36***

Overview

You use the same steps to upgrade QAD Configurator from the following previous versions of QAD Configurator to the current version:

QAD Configurator version 4.3, 4.3.1, 4.4, 4.4.1, 4.4.2, 5.0.1, 5.0.2, 5.1, and 5.2.

General upgrading steps are as follows:

1. Installing QAD Configurator 5.2.1
2. Copying Data From the Previous Version of QAD Configurator
3. Updating Database Schema
4. Converting Production Database to UTF-8
5. Converting Production Database to the Current Version and Loading Production Data
6. Rebuilding Database Index
7. Converting Production Database to Enterprise Edition
8. Modifying the manifest.qpkg File

Prior to Upgrading QAD Configurator

If your existing QAD Configurator system uses an earlier Progress version, upgrade it to OpenEdge10.1A02 or up first. See Progress documentation for information on how to upgrade Progress.

QAD Configurator works with the following versions of QAD Enterprise Applications:

eB2.1 SP4, QAD 2007, QAD 2007.1, QAD 2008 Standard, QAD 2008.1 Standard, QAD 2008.1 Enterprise, QAD 2009 Standard, QAD 2009 Enterprise, QAD 2009.1 Enterprise, QAD 2010 Enterprise, and QAD 2010.1 Enterprise

If you are using previous versions of the QAD Enterprise Applications, you must first upgrade to a supported version. See the installation documentation for that version for information.

Back up your QAD Configurator production environment prior to upgrading.

Upgrading Steps

Installing QAD Configurator 5.2.1

Install a new instance of QAD Configurator 5.2.1. See “Installing QAD Configurator” on page 1 for information on how to install QAD Configurator 5.2.1.

Copying Data From the Previous Version of QAD Configurator

1. Remove the production databases from the new QAD Configurator 5.2.1 database installation directory.
2. Copy the legacy production databases of the previous version of QAD Configurator to the QAD Configurator 5.2.1 database installation directory.

Updating Database Schema

Use the following steps to load the new data definitions to update the QAD Configurator database schema in the new Configurator instance.

1. Launch MFG/UTIL for Configurator. Execute `cpdutil` from *`CfgDBSvrInstallDir`*`/cpd/`.

   Note MFG/UTIL assumes that the QAD Enterprise Applications database uses the utf-8 code page. If your QAD Enterprise Applications database uses a different code page, open `cpdutil` in a text editor and modify the startup parameters -cpinternal, -cpstream, -cpcoll to use the same code page used by your QAD Enterprise Applications database.

   A warning message may appear telling you that the code page you use is not supported for TTY clients and that you may corrupt files or databases. You can safely ignore it.

2. The MFG/UTIL screen displays. Choose Upgrade Configurator|Load Database Schema (.df) File. The Connect Database screen displays.
3. Connect to the Configurator production database.
4. Load the data definition file to update the database schema. Load the appropriate schema definition files depending on your upgrade path.

If you are upgrading from this version	**Load ...**
4.3	43-521.df
4.3.1	431-521.df
4.4	44-521.df
4.4.1	441-521.df
4.4.2	442-521.df
5.0	50-521.df
5.0.1	501-521.df
5.0.2	502-521.df
5.1	51-521.df
5.2	520-521.df

5. When loading is complete, you return to the MFG/UTIL main screen. The QAD Configurator database schema is updated.

Converting Production Database to UTF-8

Perform this step only if your QAD Enterprise Applications database also uses the UTF-8 code page.

1 In Configurator MFG/UTIL, choose Upgrade Configurator|Convert Configurator to UTF-8.

2 Choose Database|Connect. The Connect Database screen displays. Connect to the Configurator production database.

3 A message appears reminding you to back up the database. Since you already have a backup copy of the legacy Configurator production database, enter Yes to continue.

4 When the Load Data Definition File screen displays, load *ProgressInstallDir*`/prolang/utf/ICU-UCA.df`.

Note Safely ignore any error and warning messages and choose OK to continue.

5 A database conversion progress screen displays. Choose Close.

6 A Rebuild Index Progress screen appears. Choose Close. The database conversion is complete.

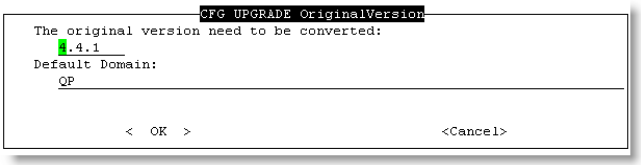
Converting Production Database to the Current Version and Loading Production Data

Perform this step only if you are upgrading from Configurator version 4.3, 4.3.1, 4.4., 4.4.1, or 4.4.2.

1 In Configurator MFG/UTIL, choose Upgrade Configurator|Convert Configurator.

2 Enter 4.3.1 for both Configurator 4.3 and 4.3.1; enter 4.4 for Configurator 4.4, 4.4.1 for Configurator 4.4.1, and 4.4.2 for Configurator 4.4.2; then choose OK.

Fig. 2.1
Convert Configurator Database



3 The Connect Database screen displays. Connect to the Configurator production database; then press OK.

4 The Configurator production database is converted.

5 In Configurator MFG/UTIL, choose Database|Progress Data Dictionary.

6 The Connect Database screen displays. Connect to the Configurator production database; then press OK.

7 Choose Admin|Load Data and Definitions|Table Contents.

8 In the Select Tables screen, choose Select Some and enter * in the Table Name field; then choose OK to select all the tables in the database.

9 Back in the Select Tables screen, choose OK.

10 In the Load Data Contents for All Tables screen, enter *CfgDBSvrInstallDir*/cpd/convdata; then choose OK.

11 The data load progress begins.

Rebuilding Database Index

1 In Configurator MFG/UTIL, choose Upgrade Configurator|Rebuild Index.

2 The Connection Database screen appears. Connect to the Configurator production database and choose OK.

3 The rebuild index progress begins. When rebuilding is complete, choose Close.

Converting Administration Database

Perform this step only if you are upgrading from Configurator version 5.0, 5.0.1, 5.0.2, 5.1, or 5.2.

1 In Configurator MFG/UTIL, choose Upgrade Configurator|Convert QAD Admin Database.

2 The Connect Database screen appears. Connect to the QAD Admin database and choose OK. Leave mfgprod as the logical name.

3 The conversion progress screen appears. When conversion is complete, choose Close.

4 The Input Directory screen appears. Browse to *CfgDBSvrInstallDir*/cpd/data/adm and choose OK.

5 The Table Selection for Load screen appears. Make sure all tables are selected and choose OK.

6 The data loading screen appears. When loading is complete, choose Close.

Converting Production Database to Enterprise Edition

Perform this step only if you are upgrading from Standard Edition or previous versions of QAD Enterprise Applications to Enterprise Edition.

1 In Configurator MFG/UTIL, choose Upgrade Configurator|Convert to eB3.

2 The Connect Database screen displays. Connect to the Configurator production database and QAD Enterprise Applications database; then press OK.

3 The conversion progress screen displays. When conversion is complete, choose Close.

Modifying the manifest.qpkg File

If you are upgrading from 4.4, 4.4.1, 4.4.2 or 5.x to a later version running on .NET UI 2.7.1 or later, you need to modify the following file:

TomcatInstallDir/webapps/qadhome/packages/plugins/ manifest.qpkg

Locate the following line in the file and increment the last digit of the version number by one; for example, if *VersionNumber* is 2.7.216.3, change it to 2.7.216.4.

```
<package confirm="true" id="qad.plugins" manifest-file="manifest.qpkg" name=
"qad.plugins" path-from-repository="plugins" version="VersionNumber">
```

Appendix A

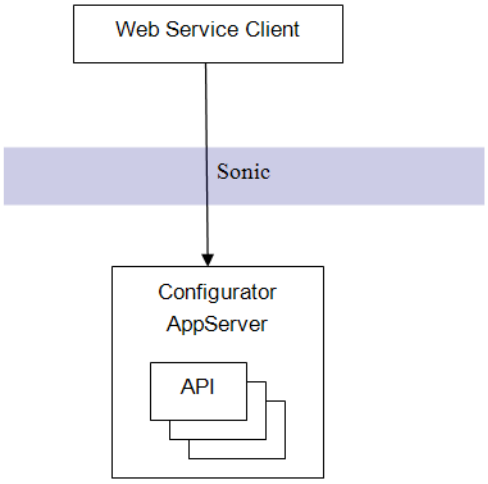
Configurator Web Service APIs

This section describes how to use the Configurator Web service APIs.

Configurator Web Service APIs Overview

A high level view of how the Web service APIs work is shown in the following diagram:

Fig. A.1
QAD Configurator Web Service APIs



The APIs are exposed through request XML documents that are processed through a SOAP Web service and return the resulting response XML documents. Creation of the request XML document and processing of the resulting response XML document are the responsibility of the Web service client.

Session Context

In order to make a Web service call, it is necessary to establish the session context for the call. In the submitted XML, the session context is set up as a collection of ttContext records. An example of a ttContext record for setting the session's domain value is shown below (all QAD ttContext properties have a qualifier value of QAD):

```
<dsSessionContext>
...
<ttContext>
<propertyQualifier>QAD</propertyQualifier>
<propertyName>domain</propertyName>
<propertyValue>st92bmfg</propertyValue>
</ttContext>
...
</dsSessionContext>
```

The session context is specified in the ttContext records as described in the following table:

Property Name	Description	Value
datasetInName	Name of input dataset for this API	Optional—typically left blank
datasetInOutName	Name of input-output dataset for this API	Optional—typically used
datasetOutName	Name of output dataset for this API	Optional—typically left blank

Property Name	Description	Value
domain	QAD Enterprise Applications domain to log into when processing this API	Required
methodName	Name of API method to be called	Required
programName	Name of Progress program in which API method resides	Required
username	User ID for logging into QAD application which provides API	Required
password	Password for logging into QAD application which provides API	Required
sessionID	Token used to identify a previously established QAD Enterprise Applications session. An invalid or missing token will cause authentication to take place through the provided username and password values.	Optional—since this token will not be used to reuse a previously established session, this property can either be excluded or have its value set to blank.
functionalAck	Specifies whether a functional acknowledgment should be returned	In those cases where a functional acknowledgment is expected to be returned this value must be set to true. If a normal Web service response is expected then this value should be set to false or not provided (defaults to false).
oneWaySend	Determines whether the functional acknowledgment is returned to the reply to address (if set to true) or returned to the blocked caller (if set to false or not specified). This value should either be set to false or not provided.	Optional—must be either set to false or not provided
replyToProgramName	This value is used only by Progress Service Interface clients to name the Progress program that will process the functional acknowledgment	Optional—not used since Avery does not use asynchronous API calls

Property Name	Description	Value
serviceInstance	This value is used only by Progress Service Interface clients to resolve which Web service to call based on settings provided in client's ServiceConnection.xml file.	Optional—only used for Progress Web service clients, such as implemented for determining sourcing.
shutdownSession	Determines whether session context created for API call gets cleaned up after completing call.	Required—must be set to true in order to properly manage API sessions it is necessary to make sure all session records are removed when API call completes.

A full sample dsSessionContext is provided below:

```
<dsSessionContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>datasetInName</propertyName>
   <propertyValue/>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>datasetInOutName</propertyName>
   <propertyValue>dsSalesOrder</propertyValue>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>datasetOutName</propertyName>
   <propertyValue/>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>domain</propertyName>
   <propertyValue>st92bmfg</propertyValue>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>functionalAck</propertyName>
   <propertyValue>true</propertyValue>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>methodName</propertyName>
   <propertyValue>callChange</propertyValue>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>shutdownSession</propertyName>
   <propertyValue>true</propertyValue>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>oneWaySend</propertyName>
   <propertyValue>false</propertyValue>
</ttContext>
<ttContext>
   <propertyQualifier>QAD</propertyQualifier>
   <propertyName>password</propertyName>
   <propertyValue>idkdiqnsdljajbdc</propertyValue>
</ttContext>
<ttContext>
```

```
      <propertyQualifier>QAD</propertyQualifier>
      <propertyName>programName</propertyName>
      <propertyValue>edixch.p</propertyValue>
   </ttContext>
   <ttContext>
      <propertyQualifier>QAD</propertyQualifier>
      <propertyName>replyToProgramName</propertyName>
      <propertyValue>dom/SOServiceInterface.p</propertyValue>
   </ttContext>
   <ttContext>
      <propertyQualifier>QAD</propertyQualifier>
      <propertyName>serviceInstance</propertyName>
      <propertyValue>st92bmfg</propertyValue>
   </ttContext>
   <ttContext>
      <propertyQualifier>QAD</propertyQualifier>
      <propertyName>username</propertyName>
      <propertyValue>demo</propertyValue>
   </ttContext>
   </dsSessionContext>
```

Note that the ttContext record for specifying the oneWaySend property value (italicized above) is provided with a setting of “false.” Similar behavior would take place if this value were removed entirely (oneWaySend defaults to a value of false). Also note that the required shutdownSession and functionalAck properties are both specified with required values of “true.”

Exceptions

Any messages resulting from processing a Web service API call will be returned in the temp_err_msg records. The elements in this record are described in the following table:

Element Name	Description
tt_level	Level in program stack at which message occurred
tt_msg_context	Information about the context of the message
tt_msg_data	Data pertaining to the message
tt_msg_datetime	Date/time at which message was created
tt_msg_desc	Message text
tt_msg_field	Field to which message applies
tt_msg_index	Used to specify where to place replacement values in message
tt_msg_keys	Comma separated list of replacement values for message
tt_msg_nbr	Unique identifier for this message type
tt_msg_processed	Whether message has already been handled
tt_msg_sev	Message level: • 1 = informational • 2 = warning • >2 = error

Messages of type informational or warning can be ignored, but error messages need to be logged by the Web service client and should be addressed immediately.

Configurator APIs

The configurator APIs were developed to allow the user to programmatically interact with the QAD Configurator.

The configurator APIs are defined below.

Retrieve Configuration Groups

The getGroups API is used to retrieve Configurator groups for a specified configurable item. This API is used by the client to determine whether an item is configurable (has one or more groups associated with the configurable item) and provides a list of groups from which the desired group is selected for further processing.

The resulting ttItemGroup records will hold the data for the Configurator groups, if any.

Retrieve Configuration Groups Example

To retrieve Configurator groups for a specified configurable item, a request XML document is created as shown below:

```
...
<ttItemGroup>
   <itemNumber>SPEC</itemNumber>
</ttItemGroup>
...
```

The resulting XML will contain the resulting groups in the ttItemGroup record. A sample of the response XML is provided below:

```
...
<ttItemGroup>
   <itemNumber>spec</itemNumber>
   <itemGroup>CRI01</itemGroup>
</ttItemGroup>
...
```

Create a New Variant Item

The createVariantItem API is used to create a new variant item from a provided configuration. Since creation of a variant item may result in creation of multiple variant items, in the case of multi-level generic items, multiple ttVariantItem records will result for the multi-level case. This API will be responsible for creating the following data in QAD Enterprise Applications: variant item, bill of materials (BOM), routings and costs.

The resulting ttVariantItem records will hold the data for the created resulting variant items for this configuration. The ttVariantItem record will contain the name of the resulting variant item, as well as the variant item's parent variant item if it is a child variant item for a multi-level generic item.

Create a New Variant Item Example

To create a variant item from a specified configuration a request XML document is created as shown below:

```
...
<dsConfigurationInput>
<ttGenericItem>
   <domain>st92bmfg</domain>
   <groupID>CRI01</groupID>
   <genericItemID>SPEC</genericItemID>
</ttGenericItem>
</dsConfigurationInput>
<dsConfiguration>
```

```
<ttConfiguration>
   <customer>4000</customer>
</ttConfiguration>
<ttConfigurationDetail>
   <feaID>siItemDescr</feaID>
   <feaExtIdx>1</feaExtIdx>
   <feaValue>Test SPEC</feaValue>
   <feaIsKey>true</feaIsKey>
</ttConfigurationDetail>
<ttConfigurationDetail>
   <feaID>siItemUOM</feaID>
   <feaExtIdx>1</feaExtIdx>
   <feaValue>EA</feaValue>
   <feaIsKey>true</feaIsKey>
</ttConfigurationDetail>
<ttPaging/>
</dsConfiguration>
...
```

The resulting XML will contain both the input records and the resulting configuration in the ttConfiguration record and the variant item in the ttVariantItem record. A sample of the response XML, which includes multi-level variant items, is provided below:

```
...
<ttVariantItem>
<variantItemID>RR005</variantItemID>
<variantItemQty>0</variantItemQty>
<variantItemPrice>0.0</variantItemPrice>
<variantItemParent/>
</ttVariantItem>
<ttVariantItem>
<variantItemID>RR005F</variantItemID>
<variantItemQty>1</variantItemQty>
<variantItemPrice>0.0</variantItemPrice>
<variantItemParent>RR005</variantItemParent>
</ttVariantItem>
<ttVariantItem>
<variantItemID>RR005P</variantItemID>
<variantItemQty>1</variantItemQty>
<variantItemPrice>0.0</variantItemPrice>
<variantItemParent>RR005F</variantItemParent>
</ttVariantItem>
...
```

Retrieve Configurations

The getConfigurationsForGenericItemByCriteria API is used to retrieve configurations for a specified generic item. This API allows for assigning criteria for filtering the configurations that are returned.

The resulting ttConfiguration records will contain all the configurations which match the provided criteria. These configurations will include the variant items, if any, associated with these configurations as well as the list price for the configurations. In addition, the ttConfigurationDetail records will have the feature option selections specified for these resulting configurations.

Retrieve Configurations Example

To retrieve configurations for specified filter criteria a request XML document is created as shown below:

```
...
<dsConfigurationInput>
<ttGenericItem>
   <domain>st92bmfg</domain>
```

```
      <groupID>CRI01</groupID>
      <genericItemID>SPEC</genericItemID>
   </ttGenericItem>
    <ttCriteria>
      <criteriaType>fea</criteriaType>
      <feaID>siItemDescr</feaID>
      <feaExtIdx>1</feaExtIdx>
      <feaDataType></feaDataType>
      <feaValue>Test SPEC</feaValue>
      <operator>eq</operator>
   </ttCriteria>
    <ttCriteria>
      <criteriaType>fea</criteriaType>
      <feaID>siItemUOM</feaID>
      <feaExtIdx>1</feaExtIdx>
      <feaDataType></feaDataType>
      <feaValue>EA</feaValue>
      <operator>eq</operator>
   </ttCriteria>
   </dsConfigurationInput>
   <dsConfiguration>
   <ttGenericItem>
      <domain>st92bmfg</domain>
      <groupID>CRI01</groupID>
      <genericItemID>SPEC</genericItemID>
   </ttGenericItem>
   <ttCriteria>
      <criteriaType>fea</criteriaType>
      <feaID>siItemDescr</feaID>
      <feaExtIdx>1</feaExtIdx>
      <feaDataType></feaDataType>
      <feaValue>Test SPEC</feaValue>
      <operator>eq</operator>
   </ttCriteria>
   <ttCriteria>
      <criteriaType>fea</criteriaType>
      <feaID>siItemUOM</feaID>
      <feaExtIdx>1</feaExtIdx>
      <feaDataType></feaDataType>
      <feaValue>EA</feaValue>
      <operator>eq</operator>
   </ttCriteria>
   </dsConfiguration>
   ...
```

The resulting XML will contain both the input records and the resulting configurations in the ttConfiguration and ttConfigurationDetail records. A sample of the response XML, with the input records removed is provided below:

```
   ...
   <ttConfiguration>
   <resultID>00000064</resultID>
   <variantItem>RR004</variantItem>
   <customer>00010000</customer>
   <confStatus>F</confStatus>
   <listPrice>0.0</listPrice>
   <netPrice>0.0</netPrice>
   <confDesc xsi:nil="true"/>
   <confComment xsi:nil="true"/>
   <createDate>2008-03-05</createDate>
   <createTime>08:26:45</createTime>
   <confUser>mfg</confUser>
   <ttConfigurationDetail>
      <resultID>00000064</resultID>
      <feaID>siItemDescr</feaID>
      <feaExtIdx>1</feaExtIdx>
      <feaValue>Test SPEC</feaValue>
      <feaListPrice>10.0</feaListPrice>
      <feaNetPrice>5.0</feaNetPrice>
      <feaManualList/>
      <feaIsKey>false</feaIsKey>
```

```
</ttConfigurationDetail>
<ttConfigurationDetail>
   <resultID>00000064</resultID>
   <feaID>siItemUoM</feaID>
   <feaExtIdx>1</feaExtIdx>
   <feaValue>EA</feaValue>
   <feaListPrice>10.0</feaListPrice>
   <feaNetPrice>5.0</feaNetPrice>
   <feaManualList/>
   <feaIsKey>false</feaIsKey>
</ttConfigurationDetail>
</ttConfiguration>
<ttPaging>
<totalPage>-1</totalPage>
<pageNumber>-1</pageNumber>
<perPage>-1</perPage>
<totalRecord>1</totalRecord>
</ttPaging>
...
```

Find or Create Variant Item

This API combines the previously described getConfigurationsForGenericItemByCriteria and createVariantItem APIs into a single API, which will attempt to locate an existing configuration that matches the specified configuration. This API is called similar to the createVariantItem API.

Find or Create a Variant Item Example

To search for an existing matching configuration that already has a variant item specified or create a new variant item (if a matching configuration with a variant item was not found), a request XML document is created as shown below:

```
...
<dsConfigurationInput>
<ttGenericItem>
   <domain>st92bmfg</domain>
   <groupID>March</groupID>
   <genericItemID>32-101</genericItemID>
   <customerID>4000</customerID>
</ttGenericItem>
</dsConfigurationInput>
<dsConfiguration>
<ttConfiguration>
   <customer>4000</customer>
</ttConfiguration>
<ttConfigurationDetail>
   <feaID>housing-length</feaID>
   <feaExtIdx>1</feaExtIdx>
   <feaValue>40</feaValue>
   <feaIsKey>true</feaIsKey>
</ttConfigurationDetail>
<ttConfigurationDetail>
   <feaID>housing-height</feaID>
   <feaExtIdx>1</feaExtIdx>
   <feaValue>54</feaValue>
   <feaIsKey>true</feaIsKey>
</ttConfigurationDetail>
 <ttConfigurationDetail>
   <feaID>housing-width</feaID>
   <feaExtIdx>1</feaExtIdx>
   <feaValue>71</feaValue>
   <feaIsKey>true</feaIsKey>
</ttConfigurationDetail>
<ttConfigurationDetail>
   <feaID>paint-housing</feaID>
   <feaExtIdx>1</feaExtIdx>
```

```
      <feaValue>yes</feaValue>
      <feaIsKey>true</feaIsKey>
   </ttConfigurationDetail>
   <ttConfigurationDetail>
      <feaID>backup</feaID>
      <feaExtIdx>1</feaExtIdx>
      <feaValue>yes</feaValue>
      <feaIsKey>true</feaIsKey>
   </ttConfigurationDetail>
   <ttPaging/>
   </dsConfiguration>
   ...
```

The resulting XML will contain both the input records and the resulting variant item in the ttVariantItem records. A sample of the response XML, with the input records removed is provided below:

```
...
<ttVariantItem>
<variantItemID>32-101-013</variantItemID>
<variantItemQty>0</variantItemQty>
<variantItemPrice>0.0</variantItemPrice>
<variantItemParent/>
</ttVariantItem>
...
```

Appendix B

Troubleshooting

If you encounter errors during installation, you can go to QAD KnowledgeBase for solutions. This section lists a number of KnowledgeBase entries for your reference.

Solution	AppServer Start Failure with Invalid Version Error
ID	qad66260
Fact	Running with a PROGRESS version other than PROGRESS 10.1B.
Symptom	The AppServer instance failed in starting. An error of “Invalid version” was found in the server log. An example of the error information is “Invalid version, 1007 (expected 17391) in object file com/qad/qra/si/RPCRequestService.r. (2888)”
Cause	A file named qra.pl is included in the Configurator deliverable. That is a library file which is compiled against PROGRESS 10.1B. That might cause problem if this library file is executed against other PROGRESS versions.
Fix	The workaround is to remove qra.pl from the Configurator PROPATH and include code compiled from QRA’s xrc directory in the PROPATH.

Solution	Error: 500 Internal Server Error
ID	qad66259
Symptom	Error “500 Internal Server Error” appears in the Questionnaire screen.
Fix	In the PROGRESS Webspeed setting, on which the Configurator is running, a parameter setting of “-weblogerror” is required. This option should be added to the corresponding section of the ubroker.properties file, which is usually located under the $DLC/properties directory.

Solution	The Configuration ID does not exist in Creating Variant Item
ID	qad65857
Fact	The Progress Webspeed part of Configurator is installed on Tomcat, not Apache.
Symptom	An error of “The Configuration ID does not exist. (13191)” is reported, when trying to create a new variant item.
Cause	In Configurator, the communication between client server is via HTTP Request and Response of XML files. However, some of the XML files in transit are quite large and therefore will be segmented by the Web server in the transfer process. The integration with Tomcat and Progress Webspeed might break down when such segmentation happens, therefore the request from the client is broken, which might cause error 13191.
Fix	We recommended that you install the Progress Webspeed part of Configurator on Apache rather than Tomcat to solve the issue.

Solution	Net UI Application cannot be launched
Symptom	Error message “The caller’s temp-table parameter temp_err_msg does not match the target temp-table temp_err_msg.” in AppServer log.
Cause	QAD Enterprise Applications code and Desktop code have not been compiled against QXtend
Fix	Compile QAD Enterprise Applications code and Desktop code with *`QADERPInstallDir`*`/qxtend/xrc` and *`QADERPInstallDir`*`/qra/xrc` in the front of the compile PROPATH.

Solution	Cannot enter Configurator Menu Item
Symptom	When the user tries to access the Configurator menu, error message 6690 “The system is unable to authenticate you” appears. Error message “restoreSessionContext is not found” in AppServer log.

Fix	Add the following two parameters to the AppServer’s ubroker session:
	`srvrStartupProc=mfaistrt.p`
	`srvrShutdownProc=mfaishut.p`
	Make sure that *`QADERPInstallDir`*`/qxtend` is added to the AppServer’s PROPATH before *`QADERPInstallDir`*.

Solution	Desktop connection cannot be initialized

Symptom	After Configurator is installed, Desktop Connection Manager cannot start up any idle session (Green icon).
	Run script `connmgr.Demonstration` on server console, warning message appears:
	`WARNING: Using -cpinternal UTF-8 is not supported for TTY clients. You may corrupt files or databases. (11994)`
	`Press space bar to continue.`

Fix	Wrap scripts of QAD Desktop that do not support cpinternal codepage UTF-8. Remove `-cpinternal UTF-8` from the bottom line of QAD Desktop Wrap scripts `connmgr.Demonstration` and `telnet.Demonstration`.

Index

A

AppServer 4
  API files 19, 20
  Configuration 25

D

Desktop Telenet scripts 29

E

empty database 9

H

hardware requirements 3

P

process maps 27
production database 11

Q

QAD .NET UI
  .pf file 24
QAD Enterprise Applications Version 20

S

site-speciific data creation 20
software prerequisites 2
start/stop scripts 18

T

Trade Management 30

U

upgrade 36
UTF-8 38

W

WebSpeed
  Configuration 26
  settings 28