



Installation Guide

QAD Customer Self Service

(QAD CSS)

Installation Overview
Installing QAD CSS Using MFG/UTIL
Upgrading QAD CSS Using MFG/UTIL
Installing or Upgrading QAD CSS Using YAB
Upgrading from Non-YAB Version to YAB Version
Post-installation Tasks

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

This document contains trademarks owned by QAD Inc. and other companies.

Copyright ©2017 by QAD Inc.

CSS_IG_v0543.pdf/c6s/p9w

QAD Inc.

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

Contents

| | |
|--|------------|
| CSS Installation Guide Change Summary | vii |
| Chapter 1 Installation Overview | 1 |
| Introduction | 2 |
| Configuration Overview | 2 |
| Deployment Options | 4 |
| Deployment Examples | 4 |
| Prerequisites | 5 |
| Software Requirements | 6 |
| Requirements for Credit Card Processing | 6 |
| QAD CSS Installation | 6 |
| Chapter 2 Installing QAD CSS Using MFG/UTIL | 7 |
| Overview | 8 |
| Preparatory Information | 8 |
| CSS Installation Components | 9 |
| Running the Installation Script | 11 |
| Creating the QAD CSS Database | 12 |
| Set Up QAD CSS Environment Values | 13 |
| Create the QAD CSS Database | 14 |
| Generate Database Scripts | 17 |
| Starting the Database Server | 21 |
| Start the Database on UNIX | 21 |
| Start the Database on Windows | 21 |
| Setting Up WebSpeed | 24 |
| Set Up the WebSpeed Server on Windows | 24 |
| Set Up the WebSpeed Broker on UNIX | 30 |
| Configuring QAD CSS | 31 |
| Setting Up Web Servers | 32 |
| Set up a Virtual Directory for Apache | 33 |
| Set up a Virtual Directory for IIS | 33 |
| Set up a Virtual Directory for Tomcat | 33 |
| Install and Configure WebSpeed Messenger | 34 |
| Applying Patches | 36 |
| Verifying QAD CSS Setup | 37 |

| | |
|---|-----------|
| Start the Broker | 37 |
| Verify Database Connection | 38 |
| Verify PROPATH | 39 |
| Administering WebSpeed | 39 |
| Generating a WebSpeed Error File | 40 |
| Compiling QAD CSS Source Code | 41 |
| Setting Up QXtend Inbound | 42 |
| Completing Installation Setup | 45 |
| Set QAD CSS Directory Paths | 45 |
| Set the Default Data Source | 47 |
| Configure Settings in QAD CSS | 47 |
| Update QAD ERP for Credit Card Processing | 49 |
| Setting Up Integration of QAD CSS and QAD Configurator (Optional) | 51 |
| Setting Up QAD CSS for Single Sign-on (Optional) | 51 |
| Installing SAML Extension (Optional) | 52 |
| Setting Up QAD CSS for User Synchronization (Optional) | 54 |
| | |
| Chapter 3 Upgrading QAD CSS Using MFG/UTIL | 57 |
| Overview | 58 |
| Upgrading CSS | 59 |
| Upgrading Progress to OpenEdge 10.1A or Above | 59 |
| Installing New QAD CSS Media | 59 |
| Setting Up QAD CSS Environment Values | 59 |
| Generating Database Scripts | 60 |
| Converting the Database | 61 |
| Starting the Database Server | 64 |
| Modifying the QAD CSS Setup | 64 |
| Applying Patches | 64 |
| Verifying QAD CSS Setup | 64 |
| Generating a WebSpeed Error File | 64 |
| Compiling QAD CSS Source Code | 64 |
| Completing Conversion Setup | 64 |
| Final Steps | 64 |
| | |
| Chapter 4 Installing or Upgrading QAD CSS Using YAB | 65 |
| Overview | 66 |
| Installing or Upgrading QAD CSS Using YAB | 66 |
| (Optional) Modifying the YAB Configuration File | 67 |
| | |
| Chapter 5 Upgrading from Non-YAB Version to YAB Version | 69 |
| Overview | 70 |
| Upgrade Steps | 70 |

| | |
|--|-----------|
| Chapter 6 Post-installation Tasks | 71 |
| Updating Agent Properties for QAD CSS Broker | 72 |
| Appendix A Troubleshooting | 73 |
| Product Information Resources | 75 |
| Index | 77 |

CSS Installation Guide Change Summary

The following table summarizes significant differences between this document and the last published version.

| Date/Version | Description | Reference |
|---------------------------|---|---------------------|
| September 2017/CSS 5.4.3 | Updated the section Installing or Upgrading QAD CSS Using YAB | Page 65 |
| | Described how to upgrade QAD CSS from a non-YAB version to a YAB version | Page 69 |
| July 2016/CSS 5.4.2 Rev1 | Updated the section Installing QAD CSS Using YAB | Page 65 |
| March 2016/CSS 5.4.2 | Updated the section Generate Database Scripts | Page 18 |
| | Updated the section Setting up QAD CSS for Single Sign-on (Optional) | Page 51 |
| | Added a section Setting up CSS SAML (Optional) | Page 52 |
| | Introduced installing CSS product in a YAB environment | Page 65 |
| | Added an entry to the Troubleshooting section | Page 74 |
| April 2015/CSS 5.4.1-Rev1 | Updated the information on generating database scripts for QAD EE | Page 18 |
| | Added the Troubleshooting section as an appendix | Page 73 |
| March 2015/CSS 5.4.1 | Deleted the <code>css_se</code> and <code>css_ee</code> directories throughout the book | - |
| | Updated the information on Requirements for Credit Card Processing | Page 6 |
| | Updated the information in the Install Script Steps table | Page 12 |
| | Modified descriptions of the section <i>Updating QAD ERP for Credit Card Processing</i> | Page 49 |
| | Updated the information on Setting Up QAD CSS for Single Sign-on | Page 51 |
| | Added steps to set up QAD CSS for User Synchronization | Page 54 |
| September 2014/CSS 5.4 | Updated the information on setting the PROPATH for generating database scripts | Page 17 |
| | Added a section about setting up a virtual directory for Tomcat | Page 33 |
| | Updated the information on configuring the <code>verisign.ini</code> file | Page 50 |
| | Updated the information on updating version name from the Upgrade CSS menu | Page 63 |
| | Added a step to clean credit card information during database conversion | Page 61 and Page 63 |
| April 2014/CSS 5.3 | Updated OpenEdge version information | Page 6 |
| | Updated the information about generating database scripts | Page 18 |
| March 2014/CSS 5.3 | Rebranded for CSS 5.3 | - |
| | Added Single Sign-on | Page 51 |
| September 2013/CSS 5.2.3 | Rebranded for CSS 5.2.3 | - |
| | Updated descriptions for Database Conversion | Page 63 |
| March 2013/CSS 5.2.2 | Rebranded for CSS 5.2.2 | - |

| Date/Version | Description | Reference |
|--------------------------|--|-----------|
| September 2012/CSS 5.2.1 | Rebranded for CSS 5.2.1 | - |
| | Updated the section Setting Up QXtend Inbound | Page 42 |
| | Updated the conversion path figure | Page 58 |
| | Updated the section Setting Up QAD CSS Environment Values in Upgrading CSS | Page 59 |
| March 2012/CSS 5.2 | Updated the Hardware and Networking Requirements section | Page 5 |
| | Added steps to enable the integration of QAD Configurator and QAD CSS | Page 51 |
| September 2011/CSS 5.1.3 | Rebranded for CSS 5.1.3 | - |
| | Added two languages, Czech and Korean, for cpstream settings | Page 13 |
| | Changed <code>css.pf</code> to <code>qadcss.pf</code> | Page 28 |
| | Added <code>binaryUploadMaxSize=-1</code> for the sample file | Page 31 |
| | Added a step for versions earlier than 2011 EE in Configuring QAD CSS | Page 32 |
| | Updated the section Applying Patches | Page 36 |
| | Added a note for the compiling step in Compiling CSS Source Code | Page 42 |
| | Changed compiling steps for Credit Card Processing | Page 49 |
| | Added a step Generating Database Scripts in Upgrading QAD CSS | Page 58 |
| | Changed <code>51.df</code> to <code>513.df</code> | Page 62 |
| | Updated steps for CSS database conversion | Page 64 |

Installation Overview

This chapter includes an overview of the QAD Customer Self Service (QAD CSS) product architecture.

Introduction 2

Configuration Overview 2

Deployment Options 4

Prerequisites 5

QAD CSS Installation 6

Introduction

QAD Customer Self Service (QAD CSS) is a Web-based storefront application designed to work with QAD Enterprise Resource Planning (ERP) applications. It lets you extend QAD ERP to the Web in support of either business-to-business (B2B) order entry or business-to-customer (B2C) order entry.

Note Two current QAD ERP applications can be used with QAD CSS—QAD Standard Edition (SE) and QAD Enterprise Edition (EE)—as well as earlier versions branded as MFG/PRO eB2.1 or QAD 2007. Except in situations where a specific version is required, this guide refers to those ERP applications as “QAD ERP.”

QAD CSS features include:

- Supports rapid integration and implementation, using a cost-effective approach and flexible Web technology
- Lets you easily model your own business processes without invasive code changes
- Style sheets that let you easily customize the user interface
- Lets your customers enter orders through a browser that updates QAD ERP in real time
- Lets your customers check their order status, credit history, and inventory levels
- Automatically generates e-mails based on rules that you define

This guide includes instructions for installing QAD CSS on UNIX, Linux, and Windows platforms, as well as configuring the product to communicate with QAD ERP and updating it from an earlier version.

The instructions are for the QAD CSS system administrator who is installing the QAD CSS database and is familiar with the UNIX operating system, the Microsoft Windows operating system, Progress software, and networking as necessary. The process requires the system administrator be also familiar with QAD ERP installation and administration.

For details on how to implement QAD CSS, see *Implementation Guide: QAD Customer Self Service*.

For details on how to use the product in day-to-day operations, see *Administration Guide: QAD Customer Self Service*.

Configuration Overview

QAD CSS is designed as an n -tier application using Progress WebSpeed. The following technologies support WebSpeed:

- HTML
- DHTML
- XML
- JavaScript
- Progress 4GL

The product is constructed in multiple layers that can be deployed on different tiers—or platforms—based on client-server architecture and scalability requirements.

Each layer has specific responsibility for handling an aspect of QAD CSS functionality and is differentiated based on the functional services it provides and the technology used to create it. The layers are:

- User interface (UI) layer. This layer presents and collects information that interacts with the user interface. It is written using HTML, Progress SpeedScript, and JavaScript. The UI layer creates the main container for the QAD CSS application and works with the UI business rule layer to dynamically generate page content when requested.

The UI layer ensures that the JavaScript and style sheets are included. In this context, JavaScript validates the UI data entry. Additionally, the UI layer controls the overall placement of UI elements.

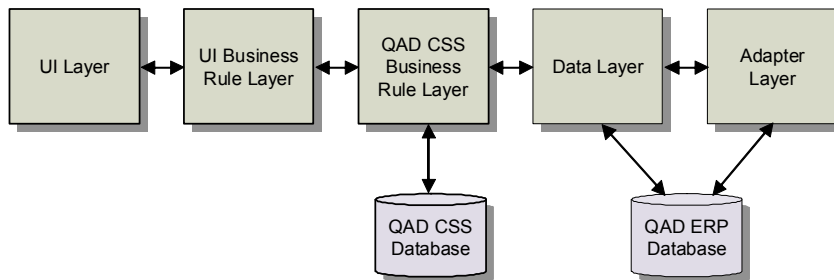
- UI business rule layer. This layer, written using WebSpeed, interacts with the UI layer and the application business rule layer (ABRL). It controls requests for data and data submission to the ABRL. It also controls dynamic creation of page content.

The UI business rule layer includes three base components:

- One that works with the UI layer to provide the base dynamic content
- A component that provides the default customizable UI generations for the product registry
- A UI extension component, where user modification to dynamic content takes place
- Application business rule layer (ABRL). The primary function of this layer, written in Progress 4GL, is to process data requests, as well as to perform QAD CSS business rule processing. It interacts with the UI business rules layer and data layer using a series of APIs that pass information and directives between the two layers. To satisfy these requests, the ABRL interacts with the data layer, adapter layer, and the QAD CSS database.
- Data layer. This layer provides a set of centralized procedures for retrieving and navigating through sets of records. The data layer also provides common procedures for updating the QAD ERP database, including the necessary change control processes required to maintain stateless applications.
- Adapter layer. This layer controls access to the QAD ERP database and code.

Figure 1.1 illustrates the QAD CSS layer architecture.

Fig. 1.1
QAD CSS Layer Architecture



Deployment Options

QAD CSS can be deployed in various configurations ranging from locally hosted to enterprise-wide options. In planning your deployment, consider these major factors:

- Your QAD ERP configuration.
- The Web server and its platform. QAD CSS can work with most major Web servers; supported operating systems include Windows, Sun Solaris, IBM AIX, HP-UX, Compaq UNIX Tru64, SCO UnixWare 7, and Linux.
- Progress WebSpeed. WebSpeed includes separate components to support single-machine or distributed implementation.
- QAD CSS. Based on such factors as performance, you can implement all the components of QAD CSS—static Web files, dynamic HTML, business rules, adapter rules, QAD CSS database—on a single machine, or spread them across several. When your environment includes a Progress AppServer, each QAD CSS component can reside on a different tier of the enterprise network.

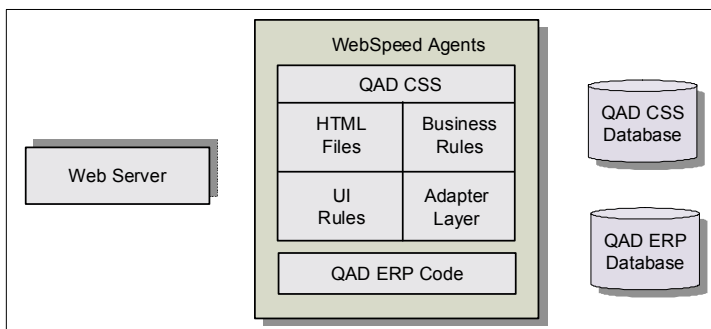
Deployment Examples

This installation guide assumes that you are installing all QAD CSS components on a single server. If you want to distribute components, this section also illustrates distributed deployment featuring Web server installation on a separate machine.

Simple Deployment

The simplest deployment of QAD CSS is to put all the components on the same server. This method is highly efficient, as well as easy to implement and maintain. However, it limits scalability and does not let you apply the kind of security typically required when users access an application over the Internet. For this reason, it is most often used to create development environments. Figure 1.2 illustrates an example of a simple deployment.

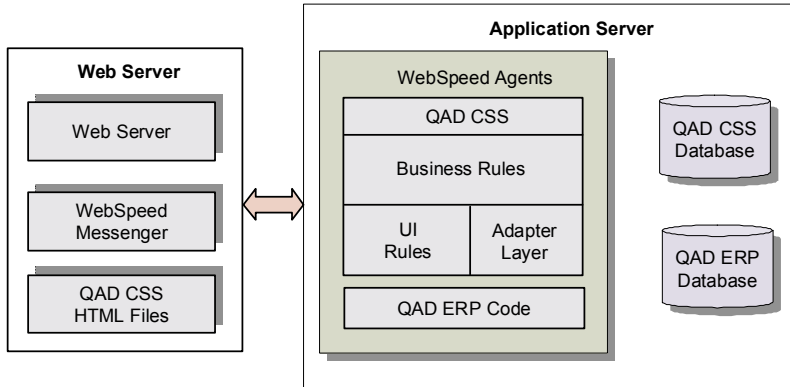
Fig. 1.2
Simple Deployment of QAD CSS



Distributed Deployment

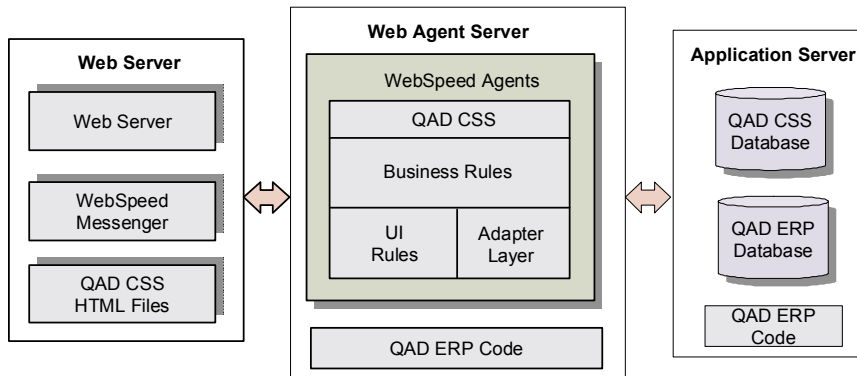
A more common deployment involves separating the Web server from the rest of the product. It is still efficient for low- to medium-load situations, and is also relatively easy to implement and maintain. Figure 1.3 illustrates an example of a distributed deployment.

Fig. 1.3
Distributed Deployment of QAD CSS



Another common example of a distributed implementation is to have the production QAD ERP database on another platform. The implementation offers more scalability and—depending on the speed and network configuration of the database server—can improve performance. Figure 1.4 illustrates an example of this type of deployment.

Fig. 1.4
Distributed Deployment of QAD CSS with Separate Database



Prerequisites

Hardware and Networking Requirements

Make sure that your system meets the following basic requirements:

- Access to the Internet, 56K dial-up at a minimum
- Support for the TCP/IP and UDP protocols. The platforms that WebSpeed supports provide built-in support for these protocols
- At least 150 MB of free space for the QAD CSS application

Software Requirements

Make sure that you have the following software elements installed and configured before you install QAD CSS:

- Compatible version of QAD ERP: eB2.1 SP2 through SP4, QAD 2007, QAD 2007.1, QAD 2008 Standard and later, QAD 2008 Enterprise and later
- OpenEdge OE10.1A and later
- Progress WebSpeed 3.1D and later

Note If you are completing a multi-tier deployment, install WebSpeed on the Web server. Alternatively, you can download and install the free WebSpeed Messenger.

- QAD QXtend, version 1.5 and later

Note CSS uses QXtend to create customers in QAD ERP.

- A Web server that supports one of the following interfaces:
 - ISAPI; for example, Microsoft Internet Information Server (IIS), version 3.x and 4.x
 - NSAPI; for example, Netscape Enterprise or Fast Track Server, Version 3.x
 - CGI 1.1; for example, Apache 2.0
- To use the browser-based editor in WebSpeed, one of the following web browsers is required:
 - Microsoft Internet Explorer 6.0 or later
 - Netscape Navigator 9.0 b3 or later
 - Firefox 2.0.0.5 or later
 - The latest version of Chrome, Opera, or Safari

Requirements for Credit Card Processing

Note that PayPal PayflowPro Gateway is used in QAD CSS 5.4 and later, so there is no need for SDK. The processing service provides the documentation required to set up the services for process transactions. (Other services may require changes to the CSS API.)

QAD CSS Installation

Installing and configuring QAD CSS requires several prerequisites.

Note See “Prerequisites” on page 5 for the prerequisite information.

The installation steps are:

- 1 Install the prerequisite components.
- 2 Install QAD CSS using the installer.

Important

- For installation with MFG/UTIL, see “Installing QAD CSS Using MFG/UTIL” on page 7.
- For Installation with YAB, see “Installing or Upgrading QAD CSS Using YAB” on page 65.

Installing QAD CSS Using MFG/UTIL

This chapter describes how to install QAD CSS using MFG/UTIL on UNIX, Linux, or Windows platforms.

| | |
|--|-----------|
| Overview | 8 |
| Preparatory Information | 8 |
| Running the Installation Script | 11 |
| Creating the QAD CSS Database | 12 |
| Starting the Database Server | 21 |
| Setting Up WebSpeed | 24 |
| Configuring QAD CSS | 31 |
| Setting Up Web Servers | 32 |
| Verifying QAD CSS Setup | 37 |
| Administering WebSpeed | 39 |
| Generating a WebSpeed Error File | 40 |
| Compiling QAD CSS Source Code | 41 |
| Applying Patches | 36 |
| Completing Installation Setup | 45 |
| Setting Up Integration of QAD CSS and QAD Configurator (Optional) | 51 |
| Setting Up QAD CSS for Single Sign-on (Optional) | 51 |
| Setting Up QAD CSS for Single Sign-on (Optional) | 51 |
| Setting Up QAD CSS for User Synchronization (Optional) | 54 |

Overview

Use the instructions in this chapter to install QAD CSS using MFG/UTIL on a UNIX, Linux, or Windows platform. MFG/UTIL is an installation and system management tool of QAD.

If you are upgrading an existing QAD CSS installation to the latest release, see “Upgrading QAD CSS Using MFG/UTIL” on page 57.

If you are installing QAD CSS on QAD EE 2016 with YAB, see “Installing or Upgrading QAD CSS Using YAB” on page 65.

Based on how you plan to use the product, you can adjust various configuration options after you begin using QAD CSS. After completing the installation and preliminary configuration process, test the new installation to verify that your QAD ERP environment communicates data correctly with QAD CSS.

Note QAD CSS is typically installed into your QAD ERP install directory.

Set Permissions (UNIX only)

For UNIX installations, make sure that the installation user has write and execute permissions for the target installation directories. QAD recommends creating a user `mfg` in the group `qad`. This process has already been done as part of the QAD ERP installation.

Preparatory Information

Before you begin the installation process, make sure that you know and write down the information about your prerequisite software components as well as your planned CSS installation as listed in Table 2.1. The table also contains naming conventions to reference directory locations that may differ for each installation.

Table 2.1
Preparatory Installation Information

| Information | Reference | Description |
|-----------------------------------|---------------------------|---|
| Progress install directory | <i>ProgressInstallDir</i> | The root directory where Progress and WebSpeed are installed on your system |
| QAD ERP install directory | <i>QADERPInstallDir</i> | The directory where QAD ERP is installed |
| QXtend Outbound install directory | <i>QXtendOutBoundDir</i> | The directory where QXtend Outbound is installed |
| Web server host or IP address | | The host name or IP address of the machine that hosts the CSS web application |
| Web server directory | <i>WebServerDir</i> | The directory where Web server software, such as Apache or IIS, is installed |
| Web server port | | A free port for your CSS Web server; by default, it is 80. |
| Source QAD ERP Domain | | The QAD ERP Domain used as the CSS data source |

| Information | Reference | Description |
|------------------------------|-------------------------|--|
| QXtend Adapter directory | <i>QXtendAdapterDir</i> | The directory where QXtend Adapter files are installed; for example, <code>/qad/qad2009/qxtend</code> This information is needed only when the QXtend Outbound events database is installed. |
| WebSpeed broker port | | A free port for the WebSpeed broker |
| QAD CSS install directory | <i>CSSInstallDir</i> | The directory where you want to install QAD CSS; for example, <code>/qad/qad2009/qadcsc</code> |
| QAD CSS database server port | | A free port for your CSS database server |
| QAD CSS product license key | | |

Note References are made throughout this document to the Progress and WebSpeed documentation. Progress documentation is available online at:

<http://www.progress.com/products/documentation/index.ssp>

CSS Installation Components

Table 2.2 lists the components included in the compressed files and summarizes their purpose.

Table 2.2
QAD CSS Components

| Directory Component | Content Description |
|-----------------------|---|
| database | QAD CSS database. |
| defs/eng | English language data definition files and data files used to create the QAD CSS database. |
| defs/xx | Non-English language data files where <i>xx</i> refers to the two-letter QAD ERP language code. |
| defs/upgrade | Data and data definition files required for upgrading from previous versions of QAD CSS, in source release-specific subdirectories. |
| demo | Data you can use to add a sample catalog to illustrate the storefront. |
| demo/defs.css | QAD CSS data. |
| demo/defs.qad | QAD ERP data. |
| demo/items | Images for the catalog. |
| qadcsc | Multiple subdirectories with core product files. |
| qadcsc/appstart.p | Additional optional startup options. |
| qadcsc/appstart.pf | Additional optional startup options. |
| qadcsc/Compme3.1.html | Default compiler program. |
| qadcsc/qadcsc.ini | QAD CSS configuration file. |

Table 2.2 — QAD CSS Components — (Page 1 of 3)

| Directory Component | Content Description |
|------------------------------------|---|
| qadcass/qadcass.pf | Default start parameters for WebSpeed agents. |
| qadcass/startcass.p | Variable setup file. |
| qadcass/ systemunavailable.html | Error file template for WebSpeed errors. |
| qadcass/css | Subdirectories with QAD CSS program files. |
| qadcass/css/ad | Administrative programs. |
| qadcass/css/adr | Administrative report programs. |
| qadcass/css/br | Browser programs. |
| qadcass/css/ca | Feedback and FAQ programs. |
| qadcass/css/cc | Credit card programs. |
| qadcass/css/cl | License clearance programs. |
| qadcass/css/ext | Extend layer programs. |
| qadcass/css/he | Help maintenance programs. |
| qadcass/css/lg | Login programs and programs used to implement the B2C order module. |
| qadcass/css/lib | Common libraries. |
| qadcass/css/messages | Error message program for serious errors. |
| qadcass/css/mfg | Programs that directly interface with QAD ERP database. |
| qadcass/css/op | Order processing programs. |
| qadcass/css/opr | Order processing report programs. |
| qadcass/css/rep | Replication templates for populating items. Note: Catalog Load and Customer Load are the preferred methods for loading this data. |
| qadcass/css/sys | System programs and base report programs. |
| qadcass/css/tools | Utility programs. |
| qadcass/css/tt | Temp table definitions used with all programs. |
| qadcass/images | QAD CSS default images. |
| qadcass/images/eng | Images for the English language. |
| qadcass/images/xx | Images for non-English languages; xx refers to the QAD ERP language code. |
| qadcass/images/items | Images for items in the catalog. Initially empty. |
| qadcass/images/oe_icons | Images for action icons. Initially empty. |
| qadcass/logs | WebSpeed server logs. Empty when created. |
| qadcass/scripts | Java scripts that are used by QAD CSS. |
| qadcass/scripts/eng_msg | English-based JavaScript messages. |
| qadcass/scripts/xx_msg | JavaScripts for non-English languages where xx refers to the QAD ERP language code. |
| qadcass/scripts/menu | Menu JavaScripts. |
| qadcass/styles | Default cascading style sheets that are used by QAD CSS. |
| qadcass/temp | QAD CSS temporary files. Empty when created. |

Table 2.2 — QAD CSS Components — (Page 2 of 3)

| Directory Component | Content Description |
|-------------------------|---|
| qadcass/temp/email | Used to process e-mail events. |
| qadcass/tools | Template and sample files. |
| qadcass/upload | Workspace for loading customers and catalogs during implementation. |
| qadcass/web | WebSpeed agent extension files. |
| qadcass/web/objects | Progress WebSpeed files. |
| qadcass/web/objects.91c | Progress WebSpeed files. |
| utils_ms | Sample scripts for use on Windows systems. |
| utils_ux | Sample scripts for use on UNIX systems. |

Table 2.2 — QAD CSS Components — (Page 3 of 3)

Running the Installation Script

- 1 Log on as a user that has permission to execute the installation script and update the installation directories.
- 2 On UNIX systems, mount the CD-ROM. On Windows, place the CD in a CD-ROM drive. Example UNIX commands are listed in Table 2.3.

Table 2.3
UNIX CD Drive Mount Commands

| Hardware | Mount Command |
|------------|--|
| Sun | <code>volcheck cdrom</code> |
| HP | <code>/etc/mount -F cdfs /dev/dsk/YourCDDevice /cdrom</code> |
| Digital | <code>mount -r -o noversion -t cdfs /dev/YourCDDevice /cdrom</code> |
| AIX | <code>smitty mountfs</code> Then select file system, directory, and file system type (<code>cdarfs</code>). |
| Linux | <code>mount /dev/hdb /mnt/cdrom</code> Where <code>/hdb</code> could be <code>hdc</code> or <code>hdd</code> among other possibilities. |
| All others | Refer to your operating system documentation or vendor for requirements to mount a CD-ROM. You can type <code>man mount</code> to determine the correct command. |

Note Copying the distribution files from the CD to a temporary directory on disk can increase extraction speed.

- 3 Launch a command window and change directories to the `install` directory on the CD:
`cd /install`
- 4 Launch the installation script in that directory:
`./install.ksh`
For Windows, launch `install.exe`.
- 5 A welcome screen displays. Press Enter. Use Table 2.4 to enter the appropriate values for script execution.

Table 2.4
Install Script Steps

| Step | User Values |
|--|---|
| License agreement | Yes |
| Install log file | Accept default or enter a new location and name |
| Create <code>instcss.ini</code> ? | Yes |
| QAD CSS install directory | The directory where you want to install QAD CSS; for example, <code>/qad/qad2009/qadcscs</code> . |
| For what language? | Simplified Chinese (ch) Castilian Spanish (cs) Czech (cz) Dutch (du) French (fr) German (ge) Italian (it) Japanese (jp) Korean (ko) Latin Spanish (ls) Polish (pl) Portuguese (po) Traditional Chinese (tw) English (us) |
| Which version of QAD Enterprise Application are you running? | 2015EE 2014EE 2013EE 2012EE 2011EE Pre2011EE SE |
| Install summary | Yes |
| File extraction | None |
| Progress directory | Your Progress install directory |
| Windows: icon folder name | Default is the latest version of CSS |
| Script end | None |

Note At the end of the script, the name and location of the installation log file display. Open the log file in a text editor to check for errors.

Creating the QAD CSS Database

QAD CSS includes the QAD installation, conversion, and configuration utility MFG/UTIL. Use this program to input several QAD CSS configuration values and to create and populate the `qadcscs` database with necessary system data.

Set Up QAD CSS Environment Values

- 1 Open the `mfgutil` script file under `CSSInstallDir` using a text editor and append the `-cpinternal` and `-cpstream` startup parameters to the last line.

For example, if the last line is:

```
$DLC/bin/_progres -p /home/envs/dlc101c/xmfgusrc/mfgutil.p -c 500 -s 63 -D 50 -TM 31 -TB
31 -B 1000 -y y 1920 -d mfy -ininame mfguprog.ini
```

Append the parameters and the last line becomes:

```
$DLC/bin/_progres -p /home/envs/dlc101c/xmfgusrc/mfgutil.p -c 500 -s 63 -D 50 -TM 31 -TB
31 -B 1000 -y y 1920 -d mfy -ininame mfguprog.ini -cpstream codepage cpinternal codepage
```

Regarding the `codepage` values of the parameters:

- For QAD SE and earlier versions, specify the same code page of your QAD ERP database as the value of both parameters; for example, for Simplified Chinese:

```
-cpinternal CP936, -cpstream CP936
```

- For QAD EE, specify the same code page of your QAD ERP database as the value of the `-cpinternal` parameter. Set the `-cpstream` parameter to the value as specified in the following table based on your system language:

| Language | cpstream |
|---------------------|-----------|
| Simplified Chinese | GB2312 |
| Traditional Chinese | BIG-5 |
| Japanese | SHIFT-JIS |
| Polish | 1250 |
| Czech | |
| Korean | EUC-KR |
| English | ISO8859-1 |
| Dutch | |
| French | |
| German | |
| Italian | |
| Latin Spanish | |
| Castilian Spanish | |
| Portuguese | |

- 2 Launch MFG/UTIL from the `CSSInstallDir` using the following command:

```
./mfgutil
```

For Windows, launch MFG/UTIL from the icon on the Start menu.

- 3 Select CSS Setup from the Configure CSS menu. Use the following screen and values to update your QAD CSS configuration.

Fig. 2.1
CSS Setup Screen

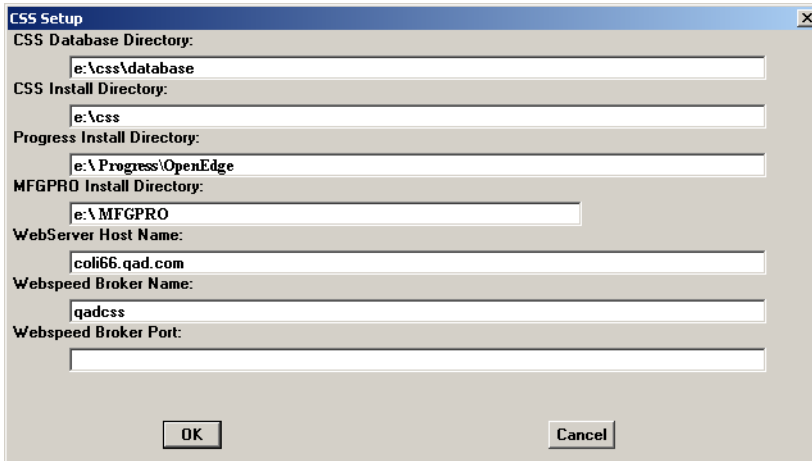


Table 2.5
CSS Setup Fields

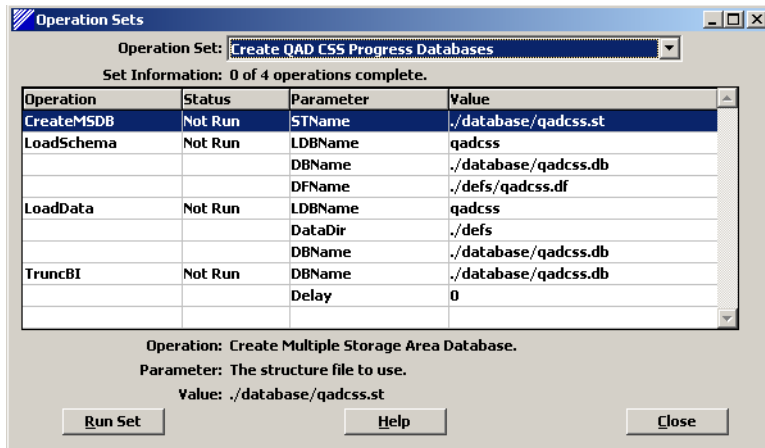
| Field | Description |
|----------------------------|---|
| CSS Database Directory | It is <i>CSSInstallDir/database</i> by default. |
| CSS Install Directory | The <i>CSSInstallDir</i> defined during install. |
| Progress Install Directory | The Progress install directory as identified during the install. |
| MFGPRO Install Directory | The <i>QADERPInstallDir</i> for the QAD ERP instance CSS connects with. |
| WebServer Host Name | The server name or IP address of the Web server. It also includes a domain name. If the domain name is not included, the data entered in the screen is not saved properly, or even lost. The <i>.qad.com</i> in the sample screen is the domain name. |
| WebSpeed Broker Name | The name of the broker you plan to use; this guide uses <i>qadcss</i> as an example. |
| WebSpeed Broker Port | An available port specified as a part of the “Preparatory Information” on page 8. |

- 4 Choose OK to save the changes.

Create the QAD CSS Database

- 1 Remaining in MFG/UTIL, select CSS Guided Setup from the Configure CSS menu.
- 2 In the Operation Set drop-down list, select Create QAD CSS Progress Database. The CSS Guided Setup steps display in the Operation screen.
- 3 Choose Run Set to start up the set.

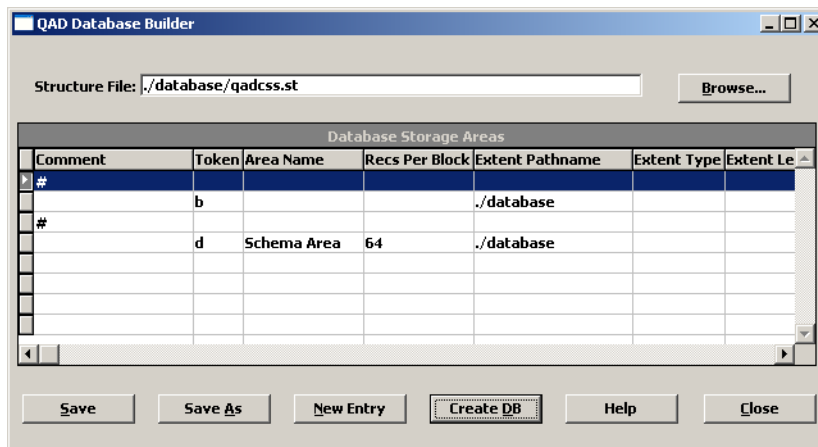
Fig. 2.2
CSS Guided Setup Screen



Note The CSS Guided Setup provides standardized steps and default values for QAD CSS database creation and the loading of schema and data. Choosing Run Set launches these steps for you. Edit the default values as you proceed if necessary.

- The QAD Database Builder opens displaying the contents of `qadcst.st`. It is the `qadcst` database structure file. You can edit the structure file in this screen.

Fig. 2.3
qadcst Structure File



- Choose Create DB to create the database.
- You are then asked what database to copy to create the database. Choose Other Database and enter the path to the empty database with the same code page of your QAD ERP databases.

Progress OpenEdge supplies a collection of empty databases in various code pages. These empty databases reside in the `ProgressInstallDir/prolang` directory in subdirectories by locale. For example, the `ProgressInstallDir/prolang/ame` directory contains empty American English databases, and the `ProgressInstallDir/prolang/sch` directory contains empty Simplified Chinese databases.

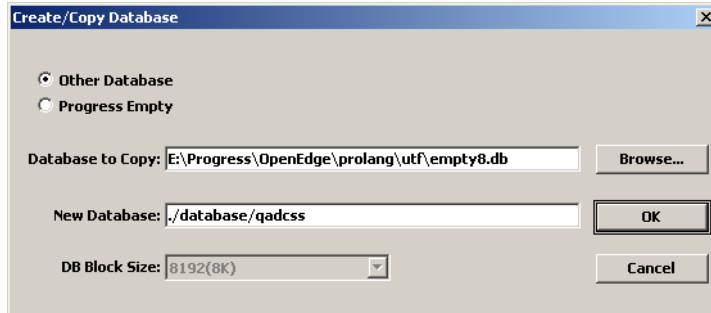
To create a database that uses the utf-8 code page, copy the following empty database:

```
ProgressInstallDir/prolang/utf/empty8.db
```

For more information on the contents of the *ProgressInstallDir/prolang* directory, see the *ProgressInstallDir/prolang/README* file.

Note Set the DB block size to 8192 (8K).

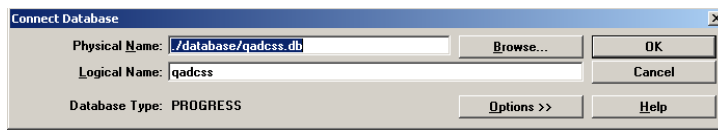
Fig. 2.4
Create/Copy the qadcss Database



7 Choose OK to create the database. A log window displays database creation progress. Choose Close to shut the log window and continue.

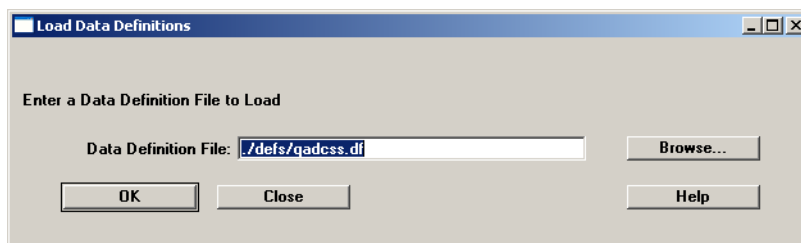
8 Once the database has been created, you are asked to connect to it.

Fig. 2.5
Connect Database



9 Choose OK to connect. The Load Data Definitions screen displays.

Fig. 2.6
qadcss Load Data Definitions Screen



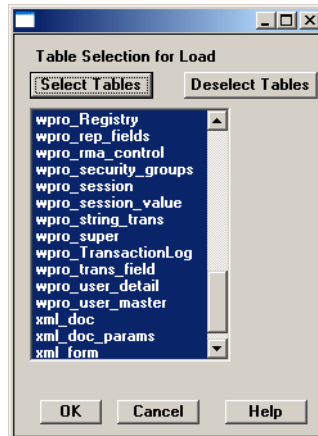
10 The correct file name and location appear by default. Choose OK. The schema loads, and a log window displays the progress. Choose Close to shut the log window and continue.

11 Choose Close in the QAD Database Builder.

12 The Connect Database screen appears again. Choose OK to reconnect to the database.

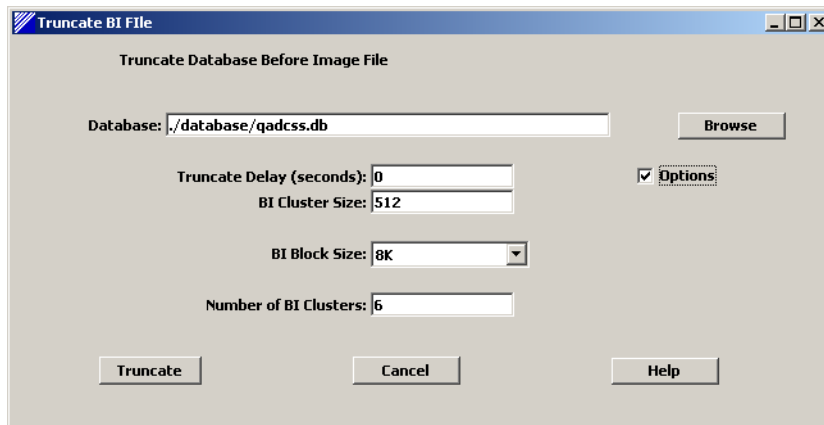
13 The Table Selection for Load screen displays. Data files exist for each table displayed. The data contained in the files is system data QAD CSS requires for operation. To start the data loads, choose OK.

Fig. 2.7
qadcss Table Selection for Load



- 14 The Log Window shows load progress. Choose Close to continue.
- 15 The Truncate BI File screen displays with correct default values. Choose Truncate to continue.

Fig. 2.8
qadcss Truncate BI File Screen



- 16 Choose Close in the log window to continue. You return to the Guided Setup window with each operation marked as Done.

Generate Database Scripts

Use the following steps to generate scripts for starting and stopping QAD CSS and QAD ERP databases.

- 1 In MFG/UTIL, choose Database Set Maintenance from the Configure CSS menu.
- 2 In the Database Set Configuration screen, select the CSS database set and choose Edit Set.
- 3 In the Database Sets window, choose Edit to set the PROPATH to the following and choose OK to save the changes:

- For QAD SE and earlier versions:

```
./
./css/mfg
```

```

./css
../EDI_API
../EDI_API/us
../mfgpro/eB21/QADERPSPVer
QXtendAdapterDir
QXtendAdapterDir/triggers
QADERPInstallDir
QADERPInstallDir/us
QADERPInstallDir/triggers
QADERPInstallDir/xrc
QADERPInstallDir/bbi
CRMInstallDir/progs
CRMInstallDir/system
CRMInstallDir/integration/mfgpro

```

Note Add *QXtendAdapterDir* and *QXtendAdapterDir/triggers* to the **PROPATH** only when installing the QXtend Outbound events database. Add *CRMInstallDir/progs*, *CRMInstallDir/System*, and *CRMInstallDir/integration/mfgpro* only when CRM is installed.

- For QAD EE:

```

./
./css/mfg
./css
QXtendAdapterDir
QXtendAdapterDir/triggers
QADERPInstallDir
QADERPInstallDir/bbi
QADERPInstallDir/fin
QADERPInstallDir/triggers
QADERPInstallDir/xrc
QXtendOutBoundDir/xrc
QXtendOutBoundDir/runtime
QADERPInstallDir/Configs
QADERPInstallDir/qra/qra.pl
CRMInstallDir/progs
CRMInstallDir/system
CRMInstallDir/integration/mfgpro
QADERPInstallDir/modules/qra-api/src

```

Note Add *QXtendAdapterDir* and *QXtendAdapterDir/triggers* to the **PROPATH** only when the QXtend Outbound events database is installed. Add *CRMInstallDir/progs*, *CRMInstallDir/system*, and *CRMInstallDir/integration/mfgpro* to the **PROPATH** only when CRM is installed.

Note Add *QADERPInstallDir/modules/qra-api/src* to the **PROPATH** for QAD 2015 EE and later.

QXtendAdapterDir. QXtend Adapter directory specified as a part of the “Preparatory Information” on page 8.

QADERPInstallDir/fin. The QAD Financials install directory that contains Financials proxy code.

../EDI_API. Contains the SO/SQ APIs required by CSS.

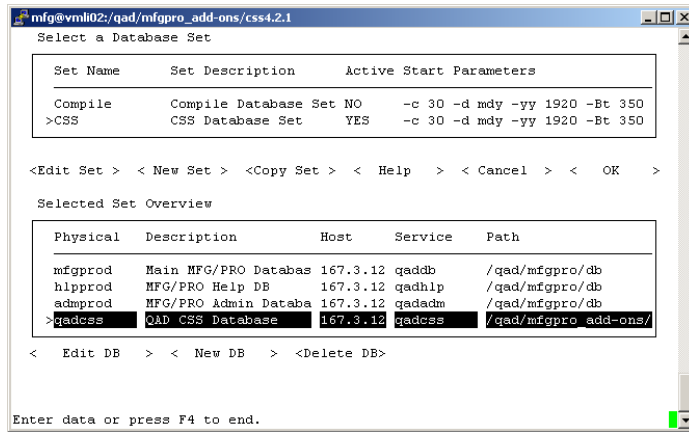
QADERPSPVer. The service pack version number of your QAD ERP; for example, SP8.

QADERPInstallDir/Configs. Contains all the required property files such as *cbserver.xml* and *verisign.ini*.

QXtendOutBoundDir. The directory where QXtend Outbound is installed.

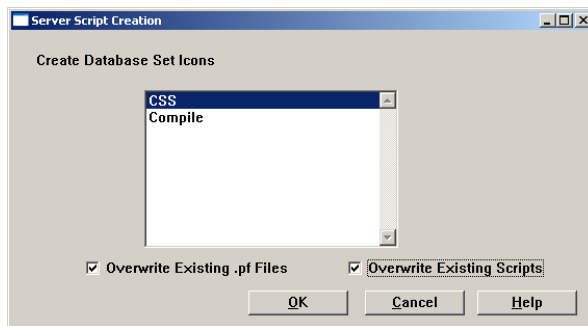
- 4 In the Database Set Configuration screen, review the configurations of the qadcss database and QAD ERP databases in the CSS database set. Choose Edit DB to modify the configurations if necessary.

Fig. 2.9
Database Set Configuration



- 5 When finished, choose OK to save the settings.
- 6 Choose Generate Scripts from the Scripts menu.
- 7 In the Server Script Creation screen, select CSS database set and choose OK.

Fig. 2.10
Server Script Creation



- 8 When prompted to confirm, Choose Yes.
- 9 When script creation is complete, choose Close.
- 10 The `start.CSS` and `stop.CSS` server scripts file are generated. For QAD SE, open and edit both files to specify the same code page of your QAD ERP database as the value of the `-cpinternal` and `-cpstream` parameters.

Note You can also manually create the database scripts. Example scripts for starting and stopping databases are included in the following directories:

- `utils_ms`: utilities formatted for use with Windows
- `utils_ux`: utilities formatted for use with UNIX and Linux

Examples of database start and stop scripts include:

- startdb

```
DLC=/qad/progress/dlc101c03; export DLC
$DLC/bin/_mprosrv /qad/qadcsc -L 1000 -B 5000 -cpinternal CP936 -cpstream CP936
```

- stopdb

```
DLC=/qad/progress/dlc101c03; export DLC
$DLC/bin/_mprshut /qad/qadcsc -by -cpinternal CP936 -cpstream CP936
```

Note When setting the code page parameters:

- For QAD SE and earlier versions, specify the same code page of your QAD ERP database as the value of both parameters; for example, for Simplified Chinese:

```
-cpinternal CP936, -cpstream CP936
```

- For QAD EE, specify the same code page of your QAD ERP database as the value of the -cpinternal parameter. Set the -cpstream parameter to the value as specified in the following table based on your system language:

| Language | cpstream |
|---------------------|-----------|
| Simplified Chinese | GB2312 |
| Traditional Chinese | BIG-5 |
| Japanese | SHIFT-JIS |
| Polish | 1250 |
| English | ISO8859-1 |
| Dutch | |
| French | |
| German | |
| Italian | |
| Latin Spanish | |
| Castilian Spanish | |
| Portuguese | |

If a TCP connection is used, QAD recommends that the port number be documented in the services file. This file is always /etc/services on UNIX systems, C:\winnt\system32\drivers\etc\services on Windows. See the server documentation for information.

Use an available port for the CSS database server specified as a part of “Preparatory Information” on page 8.

The following are example entries in the /etc/services file:

```
#EXAMPLE ENTRY:
# <service_name> <portnumber/protocol> #<description>
qadcsc 5660/tcp # QAD CSS DB
mfgprod 5661/tcp # QAD ERP Main DB
mfgadmin 5662/tcp # QAD ERP Admin DB mfghlp
```

5663/tcp

QAD ERP Help DB

Starting the Database Server

Run the QAD CSS database in multi-user mode to enable connections from multiple WebSpeed agents. Depending on the server configuration and Progress and WebSpeed versions used, a TCP connection to the database may be required.

Important Make sure that you execute this sequence successfully, and start the QAD CSS database before you continue with the installation procedure.

Start the Database on UNIX

Launch the database startup script you generated in previous steps.

The following output is produced when the database is started using the sample `start.CSS` script:

```
./start.CSS
PROGRESS Version 9.1C as of Thu Jun 7 10:03:59 EDT 2005
17:53:19 BROKER 0: Multi-user session begin. (333)
17:53:19 BROKER 0: Begin Physical Redo Phase at 320. (5326)
17:53:20 BROKER 0: Physical Redo Phase Completed at blk 435 off 3894 upd 6970. (7161)
17:53:21 BROKER 0: Started for qadcssDB using TCP, pid 24729. (5644)
```

Start the Database on Windows

Start the database servers on Windows in the Progress Explorer.

Depending on your specific Progress version, system configuration, and operating system, the Progress Explorer navigation and display layout may differ slightly from the following instructions. Refer to the Progress Explorer online help for detailed information.

- 1 Verify that the AdminServer process is running.
 - If the software is on a Windows server, open a command window and enter the following:


```
ProgressInstallDir\bin\proadsv -query
```

 If the AdminServer is not running, open Settings|Control Panel|Services from the Windows Start button. Then, select the AdminService for OpenEdge 10.1A and click Start.
 - If the software is on a UNIX server, run the following command on that server:

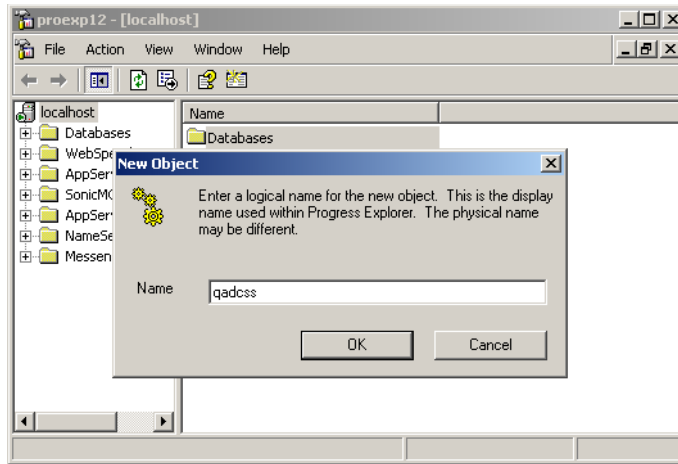

```
ProgressInstallDir\bin\proadsv -query
```

 If the AdminServer is not running, start it using the following command:


```
ProgressInstallDir\bin\proadsv -start
```
- 2 From the Windows Start menu, select Programs|Progress|Progress Explorer Tool. The localhost AdminServer and any others defined on your system display.
- 3 Verify that the correct AdminServer is on the list. (If the AdminServer is on the system from which you are running Progress Explorer, it is shown as localhost.) If the correct server is not on the list, consult the Progress documentation for instructions on adding a server service.

- 4 Right-click the server where QAD CSS is installed and select Connect. Enter the user ID and password to administer the service; by default it is the user's login ID and blank.
- 5 Right-click Databases in the right window and select New. Enter the name for the database as qadcsc and click OK.

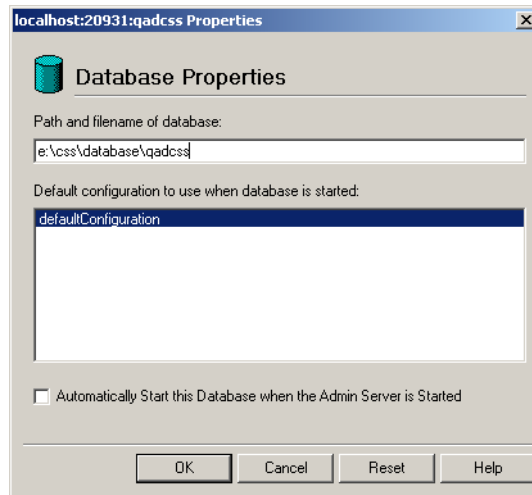
Fig. 2.11
Adding the qadcsc Database



- 6 Database Properties displays. Enter the complete path to the QAD CSS database in `CSSInstallDir/database` and click OK.

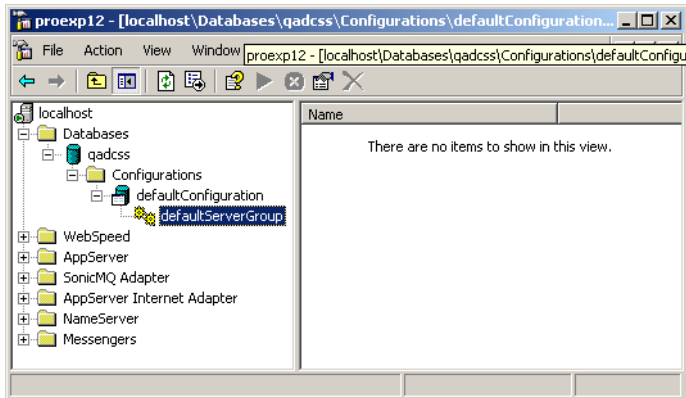
Note The `.db` extension for the database file is not required.

Fig. 2.12
Database Properties



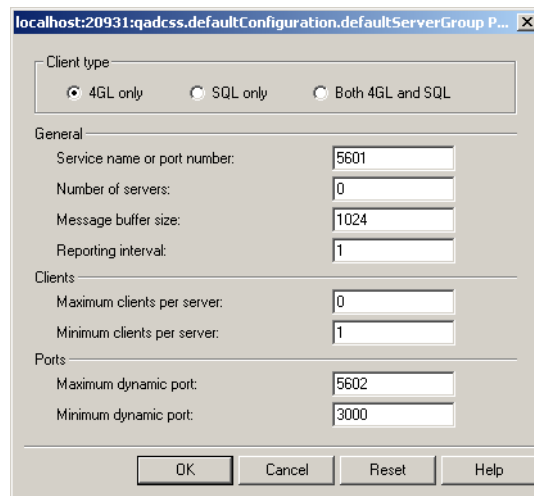
- 7 In the left window, expand the service, the database created, the configuration, and defaultConfiguration.

Fig. 2.13
Expanding Configuration Node



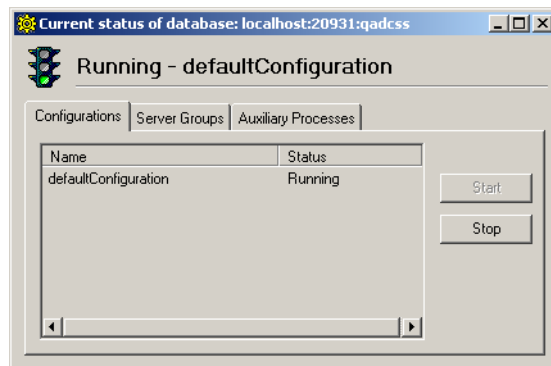
- 8 Right-click defaultServerGroup and choose Properties. In the dialog box:
 - a Select 4GL Only.
 - b Enter a free service port number in the Ports section. (If necessary, contact the system administrator for a number.)
 - c Click OK.

Fig. 2.14
Database Configuration Properties



- 9 Right-click the database name in the left window and choose Start.
- 10 Right-click the database again and choose Status. Normally the database is running.

Fig. 2.15
Database Running



Setting Up WebSpeed

This section describes how to create a WebSpeed broker for QAD CSS. If you are installing on Windows, use the Progress Explorer. If you have a Windows machine network-connected to the Unix server, also use the Progress Explorer, since the method is easier and less error-prone.

For UNIX-only systems, follow the steps in “Set Up the WebSpeed Broker on UNIX” on page 30.

This procedure assumes that you are installing all QAD CSS components on one server. In a multi-tier installation, you can install the WebSpeed messenger on the Web server.

Note Information on other WebSpeed administrative utilities is included in “Administering WebSpeed” on page 39.

Set Up the WebSpeed Server on Windows

Additional Resource: Review the “Configuring WebSpeed on Windows” chapter in the *Progress WebSpeed Installation and Configuration Guide*.

This section provides instructions for setting up the WebSpeed Server instance using the Progress Explorer, which runs on Windows machines only. You can use these instructions to set up:

- A Windows server
- A UNIX server using a network-connected Windows machine

In the following instructions, use the Progress Explorer to modify the WebSpeed section of the `ubroker.properties` configuration file. This file is located in the `properties` subdirectory below the Progress install directory.

- 1 Make sure that the AdminServer is started.
- 2 Start the Progress Explorer from the Windows Start button. The Progress Explorer window displays.

- 3 If you are configuring a remote machine, use the following instructions to add an icon for the remote machine to the Progress Explorer. If you are configuring the Progress servers for this machine, use the localhost icon.
- a Click Progress Explorer and then select Action|Add Progress Server. The Server Properties window displays.
 - b Complete the General tab in the Server Properties window using the following table as a guide:

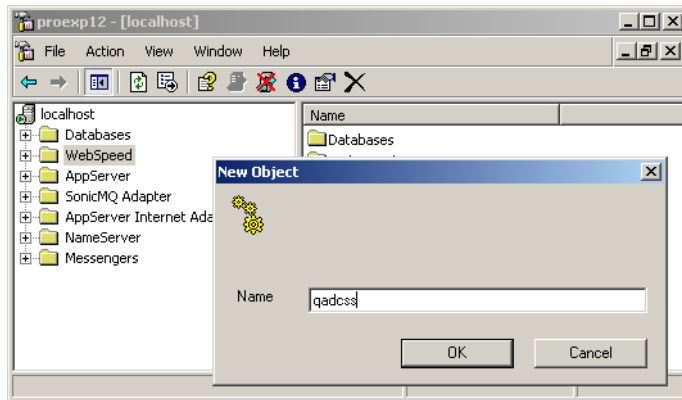
| Field | Description |
|----------|---|
| Server | Enter the machine name of the server you want to configure. |
| User | Enter the user name under which you want to log in to the remote machine. If you are configuring a remote UNIX machine, log in as a user with root-level permissions. If you are configuring a remote Windows machine, log in as a user with administrator permissions. |
| Password | Enter the password for the user that you are using to log in to the remote machine. |

- c Click OK to continue.
- 4 In the Progress Explorer window, right-click the server icon for the machine on which you want to configure the WebSpeed server instance. Choose Connect.
- 5 If you are configuring a remote machine, you are prompted for a user name and password. Use the same name and password that you specified in step 3.
- The Progress Explorer connects to the AdminServer process running on the machine where you are configuring the WebSpeed instance. A green arrow in the machine icon indicates the connection.

Set Up the WebSpeed Broker

- 1 Once you have connected to the remote or localhost server machine, click the server name. Several folders display. Right-click the WebSpeed folder and choose New.
- 2 In the New Object window, enter the name you want to use for the WebSpeed broker. The following instructions use `qadcss` as an example.

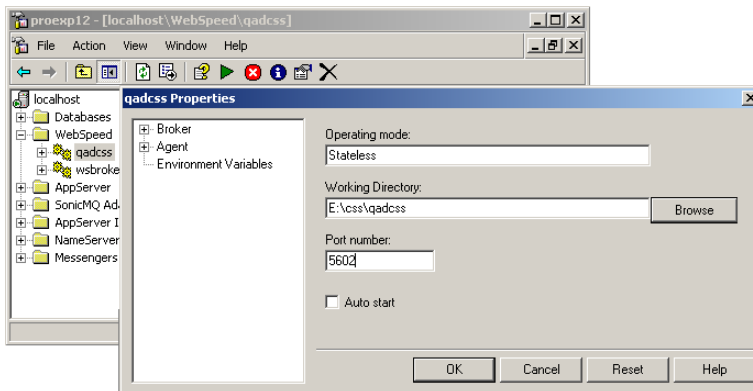
Fig. 2.16
Adding a WebSpeed Broker



When ready, click OK to continue.

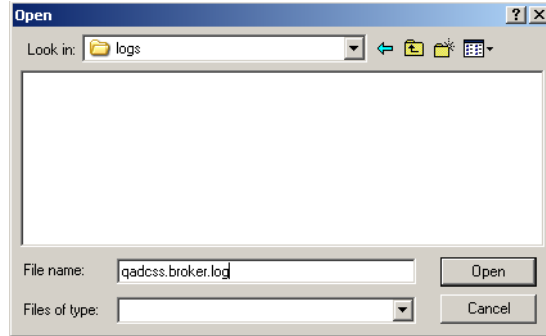
- 3 Expand the WebSpeed directory; right-click the broker that was created and choose Properties. A Properties dialog box displays.
 - a For Working Directory, click Browse and select the *CSSInstallDir\qadcss* directory.
 - b Enter a free port number to be used as the WebSpeed broker port specified as a part of the “Preparatory Information” on page 8.
 - c Leave the Operating Mode set to Stateless and Auto start set to No.

Fig. 2.17
WebSpeed Broker Properties



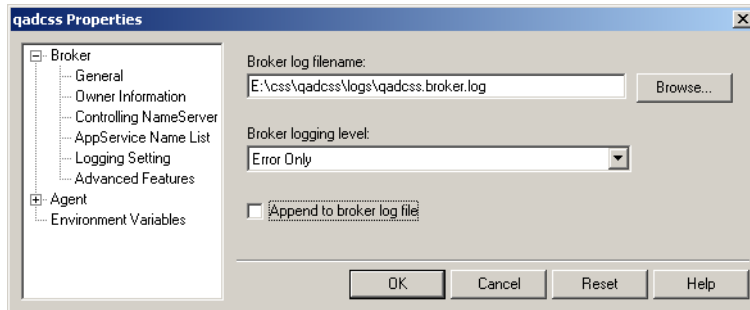
- 4 Still in the Properties screen, expand the Broker by clicking its name in the left side of the Properties window and select Logging Setting to display a new set of options:
 - a Click Browse to find the `logs` directory under the `CSSInstallDir\qadcscs`. Select the `logs` directory and in the Open dialog box, enter the file name `BrokerName.broker.log`.

Fig. 2.18
Entering the Broker Log File Name



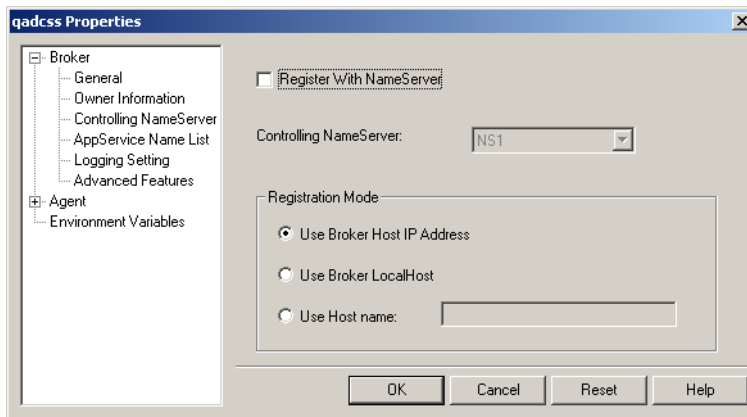
- b Click Open.
- c Choose Error Only in the Broker logging level drop-down list.
- d Uncheck Append to broker log file.

Fig. 2.19
Broker Log Settings



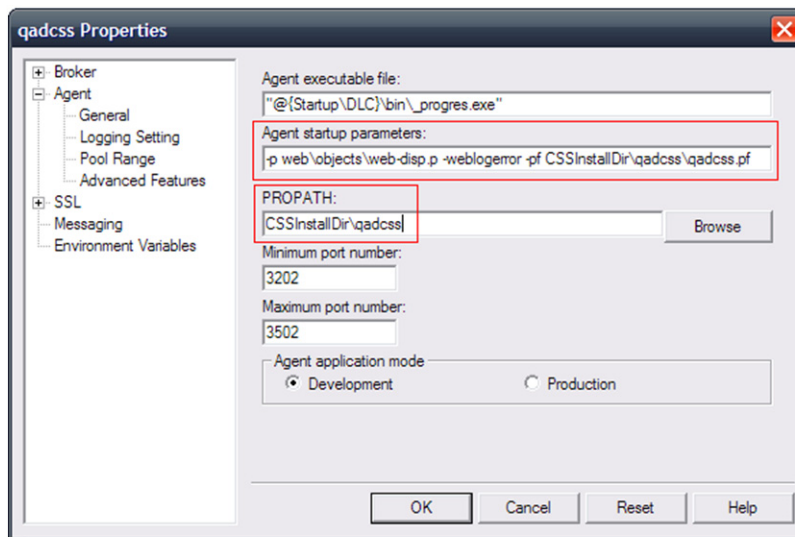
- 5 Remaining in Properties, if you are not using the NameServer, click ControllingNameServer under the Broker. In the NameServer properties, clear the Register with NameServer check box.

Fig. 2.20
Unregistering the NameServer



- 6 In Properties, expand Agent, select Logging Setting, and repeat step 4, specifying `BrokerName.server.log` as the file name.
- 7 Click General under the Agent section and set the agent startup parameter and PROPATH.

Fig. 2.21
General Agent Settings



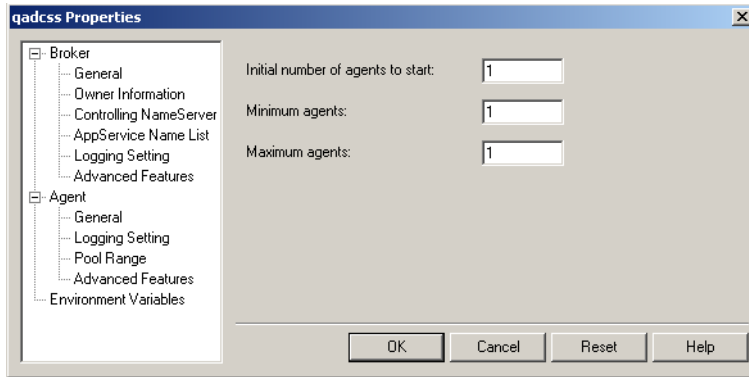
Agent startup parameters. Enter the following:

```
-p web\objects\web-disp.p -weblogerror -pf CSSInstallDir\qadcss\qadcss.pf
```

PROPATH. Enter `CSSInstallDir\qadcss`.

- 8 Click Pool Range under the Agent section. Set each value to 1 and click OK.

Fig. 2.22
Pool Range Settings

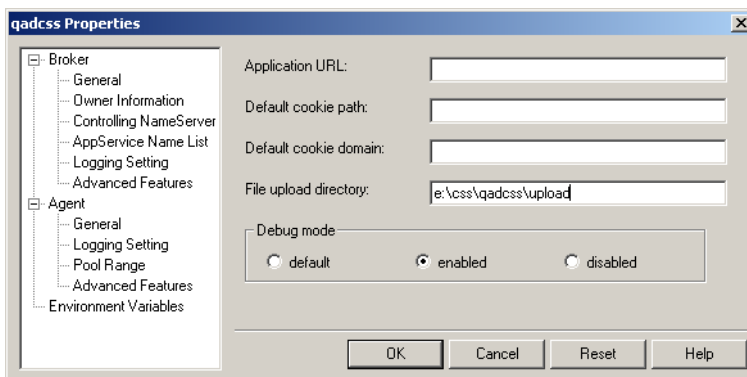


Note After you register your QAD CSS clearance code, return to this section and update the pool settings based on the licensed number of agents.

- 9 Under Agent, click Advanced Features; then set Binary upload max size to -1.
- 10 In the Properties screen, select Advanced Features under Agent, and enter the File Upload Directory. The path is:

`CSSInstallDir\qadcss\upload`

Fig. 2.23
Entering the File Upload Directory



- 11 Choose OK to complete the configuration.
- 12 Choose File|Exit to exit the Explorer tool.

Set Up the WebSpeed Broker on UNIX

The next step generates a WebSpeed broker for your QAD CSS implementation. This broker maintains the connection between your QAD CSS and QAD ERP applications. The broker definition is added to the Progress `ubroker.properties` file. The first step is to generate a sample broker definition that you can edit and append to the original file.

- 1 In MFG/UTIL, select Generate Sample `ubroker.properties` File from the Configure CSS menu. The generate screen displays.

Fig. 2.24
Sample `ubroker` Screen

- 2 Accept the defaults except for the broker port number. Enter a valid port number that is not in use.

Note Set all the agents to 1. Later, after implementation, you will need to reset the agents to the number of agents needed for your production environment.

- 3 Choose OK to generate the sample. A message displays showing the path and file name for the sample file. The file is written to:

```
CSSInstallDir\qadcscs-ubroker.properties
```

The file looks something like the following:

```
[UBroker.WS.qadcsc]
  appserviceNameList=qadcsc
  brokerLogFile=CSSInstallDir\qadcsc\logs\qadcsc.broker.log
  srvrLogFile=CSSInstallDir\qadcsc\logs\qadcsc.server.log
  description=qadcsc
  environment=qadcsc
  controllingNameServer=NS1
  portNumber=3601
  PROPATH=CSSInstallDir\qadcsc
  uuid=6fcd1c8e29b18fd3:55af5:ff79d89dd3:-8000
  workDir=CSSInstallDir\qadcsc
  fileUploadDirectory=CSSInstallDir\qadcsc\upload
  defaultService=0
  groupName=
  initialSrvrInstance=1
  maxSrvrInstance=1
  minSrvrInstance=1
  srvrAppMode=Development
  srvrDebug=Disabled
  srvrLoggingLevel=1
  srvrLogAppend=0
  brkrLogAppend=0
  brkrLoggingLevel=1
  userName=
  srvrStartupParam=-p web\objects\web-disp.p -weblogerror
  -pf CSSInstallDir\qadcsc\qadcsc.pf
  binaryUploadMaxSize=-1
  #
```

- 4 Open the sample broker file in a text editor and verify its content; edit the file if necessary. For multi-tier CSS deployment, add the following two additional lines to the end of the file:

```
RegistrationMode=Register-HostName
hostname=IPAddressofBrokerServer
```

Note Do not change the name of the sample broker file.

- 5 Make a backup copy of the original `ubroker.properties` file located in your `ProgressInstallDir\properties` directory.
- 6 In MFG/UTIL, choose Merge Sample `ubroker.properties` File from the Configure CSS menu to automatically merge the sample broker file into the Progress `ubroker.properties` file.
- 7 Restart your NameServer using the following commands.

```
nsman -i NS1 -stop
nsman -i NS1 -start
```

Configuring QAD CSS

Use these procedures to update the WebSpeed and QAD CSS startup files to support QAD CSS; then verify that the modified files work correctly.

- 1 In MFG/UTIL, choose Generate Configuration Files from the Configure CSS menu.
- 2 A log window displays showing the following startup files have been updated successfully.
 - `web-disp.p`
 - `web-util.p`
 - `qadcsc.ini`
- 3 Choose Close to close the log window.

- 4 If you have installed the QXtend database, edit *CSSInstallDir/qadcass/qadcass.ini* to add commands to connect to the QXtend database. Use the following as an example:

```
-db /dr01/qad2009/systest/db/qxevents -ld alias_qxevents  
-db /dr01/dbs/live/qxodb -ld qxodb
```

- 5 If you have installed the QAD CRM database, edit *CSSInstallDir/qadcass/qadcass.ini* to add command lines to connect to the CRM database. Use the following as an example:

```
-db /dr01/dbs/live/bisgen -ld bisgen -trig triggers  
-db /dr01/dbs/live/bisgmenu -ld bisgmenu -trig triggers  
-db /dr01/dbs/live/dataexch -ld dataexch -trig triggers
```

- 6 If you are using QAD EE, open the *CSSInstallDir/qadcass/qadcass.pf* file and check for the `-cpinternal`, `-cpstream`, and `-cpcoll` parameters. Make sure that you use the same code page and collation of the QAD ERP database; for example:

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

After generating the configuration files, stop and restart the QAD CSS service to make sure that it works correctly.

- 1 Restart the QAD CSS service (in this document, `qadcass`) by running the stop script then the start script on UNIX systems or using Progress Explorer in Windows.

Note View the *BrokerName.server.log* file in the */CSSInstallDir/qadcass/logs* directory to determine the startup status. Note any errors logged that prevented proper startup and correct the problem. If the broker started correctly, the following message displays:

```
QAD CSS: Done adding programs to the UI Super Layer.
```

- 2 Check the *CSSInstallDir/qadcass/temp* directory to see if any files were generated. If so, review the most recent files with extensions `.log` and `.log.erp` to verify that login was successful.

Setting Up Web Servers

Each Web server has different requirements. This section includes some steps for configuring the Apache web server and Microsoft's Internet Information Server (IIS). In addition, it includes steps for installing and configuring the WebSpeed Messenger, which is needed in a two-tier deployment.

In a multi-tier deployment, the */images*, */scripts*, and */styles* directories located in *CSSInstallDir/qadcass* are moved to a directory on the Web server. Create an alias for this directory.

Important In a production environment, do not alias the full *CSSInstallDir* directory because the action could create a serious security risk by making the entire source code freely available on the Web.

Set up a Virtual Directory for Apache

Follow these steps to allow the system to access QAD CSS using an Apache Web server:

- 1 Locate the `httpd.conf` file in the `WebServerDir/conf` directory.
- 2 Locate the following section in this file:

```
Alias /icons/ "WebServerDir/icons"

<Directory "WebServerDir/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow, deny
    Allow from all
</Directory>
```

- 3 Add the following lines after the code section displayed in the previous step:

```
Alias /qadcsc/ "CSSWebDir/"

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

Note In this example, `CSSWebDir` is a directory containing just the images, scripts, and styles.

- 4 Restart Apache for your changes to take effect.
For information on how to restart Apache, refer to the Apache HTTP server documentation at <http://httpd.apache.org/docs>.

Set up a Virtual Directory for IIS

Follow these steps to allow the system to access QAD CSS using Server:

- 1 From the Windows Start menu, select Settings|Control Panel.
- 2 Select Administrative Tools and then Internet Services Manager.
- 3 Right-click Default Web Site and choose New Virtual Directory.
- 4 Enter an alias name to access this Web virtual directory. Use the same naming conventions as for naming a directory; for example, `qadcsc`.
- 5 Enter the path to the directory that contains the `/images`, `/scripts`, and `/styles` subdirectories.

Set up a Virtual Directory for Tomcat

Follow these steps to allow the system to access QAD CSS using a Tomcat Web server:

- 1 Create the `CSSAlias` directory under the `<Tomcat_Home>/webapps` directory.
- 2 Create the `WEB-INF` directory under the `<Tomcat_Home>/webapps/CSSAlias` directory.

- 3 Create the `cgi` directory under `<Tomcat_Home>/webapps/CSSAlias/WEB-INF`, and copy the file `wspd_cgi.sh` from the `ProgressInstallDir/cgi-bin` directory to the `cgi` directory.
- 4 Create the `web.xml` file under the `<Tomcat_Home>/webapps/CSSAlias/WEB-INF` directory, and then edit the file to include the following information:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">

  <servlet>
    <servlet-name>cgi</servlet-name>
    <servlet-class>org.apache.catalina.servlets.CGIServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>0</param-value>
    </init-param>
    <init-param>
      <param-name>cgiPathPrefix</param-name>
      <param-value>WEB-INF/cgi</param-value>
    </init-param>
    <load-on-startup>5</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>cgi</servlet-name>
    <url-pattern>/cgi-bin/*</url-pattern>
  </servlet-mapping>
</web-app>
```

- 5 In the `context.xml` file, change `<Context>` to `<Context privileged="true">`.
- 6 Start Tomcat and confirm that the application `CSSAlias` is started in Tomcat manager.
- 7 Make sure `wspd_cgi.sh` is installed in `cgi`. If the file is not there, copy it from the `ProgressInstallDir/cgi-bin`.
- 8 Restart the Tomcat web server and WebSpeed Workshop. Here is an example of the WebSpeed Workshop URL:

```
http://WebSpeedServerHostName:8080/CSSAlias/cgi-
bin/wspd_cgi.sh/WService=WebSpeedBrokerName/workshop
```

Note 8080 is the Tomcat port on a customer system.

Install and Configure WebSpeed Messenger

If you do not have a full WebSpeed installation on your Web server in a multi-tier installation, download and install the WebSpeed Messenger to the Web server's `cgi` directory or a directory with executable permissions for scripts and executables.

The WebSpeed Messenger provides communication links between the Web server and the WebSpeed server when the Web server resides on a different server than the WebSpeed broker and agents.

Installation

Use the following steps to download and install the appropriate WebSpeed Messenger files:

- 1 Download the WebSpeed Messenger executable from the Progress Web site.
<http://www.progress.com/esd/index.ssp>
 Make sure that you download the executable for your specific server. Additionally, download any related documentation.
- 2 Review the documentation provided on the download page for additional installation or server requirements information.
- 3 Install the WebSpeed Messenger using the installation instructions provided on the download page. Note the install directory; during the configuration steps, you edit and copy several files in this directory.

Configure the WebSpeed Messenger

These steps assume that you are configuring WebSpeed on a UNIX system. If you are installing on a Windows server, these steps also apply; the relative paths are the same. The only exception is that Windows requires the `cgiip.exe` executable instead of the `wspd_cgi.ksh` executable.

- 1 If your Web server is running on a UNIX server, go to the `/WebSpeedInstallDir/properties` directory. Find `msgnrs.properties`, make a copy of this file, and rename the copy `ubroker.properties`.
- 2 For both Windows and UNIX servers, using a text editor, open the `ubroker.properties` file in `/WebSpeedInstallDir/properties`.
- 3 Find the `[NameServer.NS1]` section and add the following parameters:

```
location=remote
hostName=DataBaseServerName
```

- 4 Find the `[WebSpeed.Messengers.CGIIP]` section. Depending on the OS where your Web server is installed, add the following parameters; use `wspd_cgi.ksh` for UNIX or `cgiip.exe` for Windows.

```
msgnrExecFile=@{Startup\DLC}\bin\wspd_cgi.ksh
AllowMsgnrCmds=1
Host=DataBaseServerName
Port=NameServerPortOnDBServer
registerNameServer=1
```

- 5 Verify your edits and save the file.
- 6 Copy the WebSpeed executable from the `/dlc/bin` directory on the application server to the `cgi` directory or a directory with executable permissions for scripts and executables on the Web server.
 - For UNIX servers, copy `wspd_cgi.sh`. In the target directory, rename the file to `wspd_cgi.ksh`.
 - For Windows servers, copy the `cgiip.exe` file.
- 7 Go to the `ProgressInstallDir/properties` subdirectory on the database server. Using a text editor, open the `ubroker.properties` file for editing. Find the `[WebSpeed.Messengers.CGIIP]` section and add the following parameters:

UNIX server

```
AllowMsngCmds=0
certStorePath=@{Startup/DLC}/certs/
controllingNameServer=YourNameServer
```

Windows server

```
AllowMsngCmds=0
certStorePath=@{Startup\DLC}\certs\
controllingNameServer=YourNameServer
```

Applying Patches

For QAD 2011 EE, apply the MFG-6398 patch.

For QAD 2010 EE, apply the R1ZR patch and the MFG-6398 patch.

For QAD 2009 EE and all earlier versions of Enterprise Edition, apply the R1PM patch, the R1ZR patch, and the MFG-6398 patch. For information on the R1PM patch, see solution ID qad72463 in the QAD KnowledgeBase.

Fig. 2.25
Compile EDI_API

| Compiler Options | | R-code Destination |
|---|---|---|
| <input checked="" type="checkbox"/> Verbose | <input type="checkbox"/> Generate Compile List File | <input checked="" type="checkbox"/> Staggered QAD ERP Default |
| <input type="checkbox"/> Silent | <input checked="" type="checkbox"/> Use Existing Compile List | <input type="checkbox"/> Flat Destination |
| | | <input type="checkbox"/> Destination Same as Source |

Compile List File: /qad/mfgpro/92b/csv/EDI_API/compile.wrk <Browse>

Compile Propath: /qad/mfgpro/92b/csv/EDI_API./qad/mfgpro/92b/xrc < Edit >

Language Code: us [V] Database Set: CSS [V]

Destination Directory: /qad/mfgpro/92b/csv/EDI_API/ <Browse>

<Compile > < Help > < Close >

For QAD SE and all earlier MFG/PRO versions, use MFG/UTIL to compile the files listed in *CSSInstallDir/EDI_API/compile.wrk*.

Verifying QAD CSS Setup

Use the procedures in this section to verify that the modifications to the setup files have been done correctly.

Start the Broker

When any modifications are made to either `web-disp.p` or `qadcsc.ini`, make sure that you restart the QAD CSS WebSpeed broker.

Starting the Broker in Windows

In Windows environments, you can use Progress Explorer to stop and restart the broker. Right-click the QAD CSS service name under the WebSpeed subdirectory and choose the appropriate command.

- 1 Right-click the broker name in the left window and choose Start.
The startup process takes several minutes.
- 2 Right-click the broker name again and choose Status.
 - a On the Summary tab, confirm that the Broker Status is Active.
 - b On the Details tab, confirm that State is Available for all agents.

Starting the Broker in UNIX

Figure 2.26 illustrates the command and display sequence for restarting the WebSpeed broker in UNIX. The commands to start or restart the broker are based on the commands in the furnished example scripts, located in `CSSInstallDir/utlils_ux`. In the example, they have been copied to `CSSInstallDir`.

The WebSpeed broker requires some time to start. Monitor the startup process by performing a query. Check the server log for this WebSpeed environment for appropriate startup messages.

Note The WebSpeed agent must have a state of Available to continue with the installation.

Fig. 2.26
Sample Verification Process

```

CSSInstallDir# ./stop brokername
PROGRESS Version 9.1C as of Thu Jun 7 10:03:59 EDT 2005

Connecting to Progress AdminServer using
rmi://localhost:20931/Chimera (8280)
Searching for brokerName (8288)
Connecting to brokerName (8276)
Shut down brokerName (8277)

CSSInstallDir# ./start brokername
PROGRESS Version 9.1C as of Thu Jun 7 10:03:59 EDT 2005

Connecting to Progress AdminServer using
rmi://localhost:20931/Chimera (8280)
Searching for brokerName (8288)
Connecting to brokerName (8276)
Starting brokerName. Check status. (8296)

CSSInstallDir#
CSSInstallDir# ./status brokername
PROGRESS Version 9.1C as of Thu Jun 7 10:03:59 EDT 2005

Connecting to Progress AdminServer using
rmi://localhost:20931/Chimera (8280)
Searching for brokerName (8288)
Connecting to brokerName (8276)

Broker Name           : brokerName
Operating Mode        : Stateless
Broker Status         : ACTIVE
Broker Port           : brokerPort
Broker PID            : 2740
Active Agents         : 1
Busy Agents           : 0
Locked Agents         : 0
Active Agents         : 1
Active Clients (now, peak) : (0, 0)
Client Queue Depth (cur, max) : (0, 0)
Total Requests        : 0
Rq Wait (max, avg)    : (0 ms, 0 ms)
Rq Duration (max, avg) : (0 ms, 0 ms)

PID  State      Port  nRq   nRcvd  nSent  Started  Last Change
02680 AVAILABLE 03202 000000 000000 000000 [start] [last change]

```

Verify Database Connection

Use the following steps to access the WebSpeed Workshop and confirm that the databases are connected.

- 1 Using a Web browser, go to the WebSpeed Workshop by entering the following URL:
`http://webServer/webServerScriptsDirectory/wspd_cgi.sh/WService=brokerName/workshop`

Note The `webServerScriptsDirectory` is typically your Web server `/cgi-bin` directory.

Note In Windows environments, `cgiip.exe` is typically specified rather than `wspd_cgi.sh`.

You see the screen similar to the following figure.



- 2 Click the Databases link from the menu.

A drop-down list displays the connected databases.

Important Make sure to connect the QAD CSS and QAD ERP databases to continue with the installation.

Verify PROPATH

To update the PROPATH as part of the configuration process, first use WebSpeed Workshop to validate your changes.

- 1 From the WebSpeed Workshop, click the PROPATH menu item.
The resulting screen displays the WebSpeed PROPATH.
- 2 Confirm that these values match the entries in the `qadcsc.ini` file.

Administering WebSpeed

To help ease the task of WebSpeed/QAD CSS administration, the installation CD provides example scripts, demonstrating how to start, stop, and monitor the WebSpeed brokers and AdminServer. The files are in the following directories:

- `utils_ms`: utilities formatted for use with Windows
- `utils_ux`: utilities formatted for use with UNIX and Linux

Example scripts include:

- `startadm`

```
DLC=ProgressInstallDir; export DLC
$DLC/bin/proadsv -start
```
- `start`

```
DLC=ProgressInstallDir; export DLC
$DLC/bin/wtbman -i brokerName -start
```

- stop

```
DLC=ProgressInstallDir; export DLC
$DLC/bin/wtbman -i brokerName -stop
```
- status

```
DLC=ProgressInstallDir; export DLC
$DLC/bin/wtbman -i brokerName -query
```

For more information on testing a WebSpeed configuration, see Chapter 8.13, “Testing Your Configuration,” in the Progress *WebSpeed Installation and Configuration Guide*.

For more information on `proadsv` or `wtbman`, see Chapter 8.2.3, “WebSpeed Command-Line Utilities,” in the Progress *WebSpeed Installation and Configuration Guide*.

Generating a WebSpeed Error File

You now create a text file that redirects WebSpeed errors to a URL.

When the QAD ERP server has gone down for some reason, QAD CSS can still be used as a stand-alone product. However, when a buyer attempts to check out, a message appears that the system is currently unavailable. The text file substitutes a user-friendly message that the system is unavailable for the WebSpeed error that would normally display.

These steps create a text file, `wsCusErr.txt` in the QAD CSS install directory. After the text file is created, make sure to manually move it to the working directory used by the Progress CGI script.

- 1 Launch MFG/UTIL.
- 2 Select Generate WebSpeed Custom Error File from the Configure CSS menu. Use the following screen and values to create the custom error message.

Fig. 2.27
Generating a Custom WebSpeed Error File

The screenshot shows a dialog box titled "CSS Generate WebSpeed Custom Error File". It contains three input fields: "Error Number" with the value "0", "Error Type" with the value "2", and "URL or Message" with the value "http://coli669.qad.com/qadcss/SystemUnavailable.html". At the bottom of the dialog are "OK" and "Cancel" buttons.

Leave the Error Number set to 0 and the Error Type set to 2. The URL contains the QAD CSS host name and domain (`coli669.qad.com` in the example), the QAD CSS install directory (`qadcss`), and the `SystemUnavailable.html` page. Use the data by default.

Note The host name and domain can include the Web server port number as well, as in `coli669.qad.com:9999`.

- 3 Choose OK to save your changes. The `wsCusErr.txt` file is saved to your QAD CSS install directory.

- Copy or move the file to your Progress work directory (WRKDIR); for example, `c:\wrk` in Windows or `/dlc/wrk` on UNIX.

Compiling QAD CSS Source Code

Follow these steps to compile the QAD CSS source code from the WebSpeed Workshop:

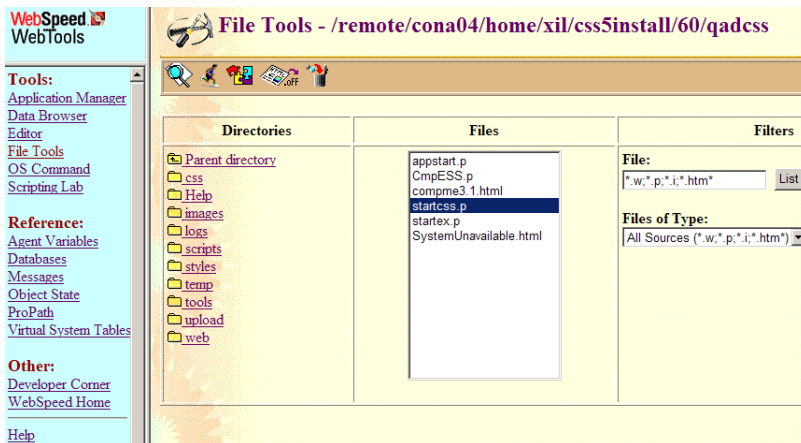
- Using a Web browser, go to the WebSpeed Workshop by entering the following URL:

`http://webServer/webServerScriptsDirectory/wspd_cgi.sh/WService=
brokerName/workshop`

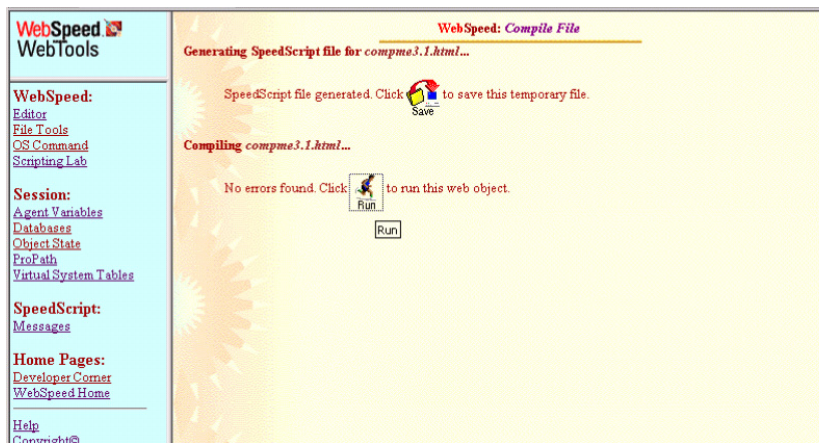
Note The `webServerScriptsDirectory` is typically your Web server `/cgi-bin` directory.

Note In Windows environments, `cgiip.exe` is typically specified rather than `wspd_cgi.sh`.

- Select File Tools from the menu on the left.
- Compile `compme3.1.html` in the WebSpeed Workshop.



- Run `compme3.1.html` from the post-compilation dialog.



- In Filter, enter `.html`. To recompile those files, highlight them first, and click Compile.

Typically, you compile the .html files only. When you are ready to move to a production environment, you can also compile the program files (.p). However, since they are loaded persistently, compiling them does not improve performance significantly.

The compilation process takes a few minutes.

Note Do not compile the .html files in the tools directory, such as rp_rpt_template.html. These files are intended to be used as samples for creating your own programs and contain code that may cause compile errors.

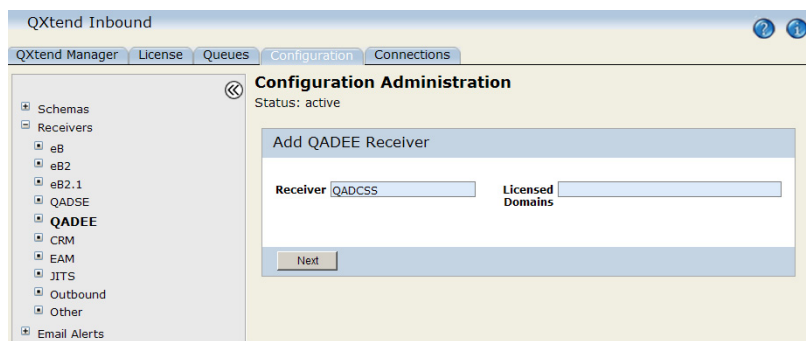
- 6 Compile the two .w files in `CSSInstallDir/qadcsc/css/cl/`. In Filter, enter .w. To recompile these files, highlight them and click Compile.

Setting Up QXtend Inbound

- 1 Make sure QXtend is correctly installed.

Note For detailed information about installing and configuring QXtend Inbound, see *Technical Reference: QAD QXtend*.

- 2 Log in to QXtend Manager.
- 3 Go to Configuration Manager and create a QADEE receiver for QAD CSS.
 - a In Configuration Manager:
 - For QAD EE, click QADEE under Receivers
 - For QAD SE and QAD 2007 versions, click QADSE under Receivers
 - For MFG/PRO eB2.1 versions, click eB2.1 under Receivers
 - b In the Configuration Administration pane, click Add.
 - c Select Continue Configuration update without suspending QXtend Inbound and click Submit.
 - d Enter a receiver name and leave Licensed Domains blank; then click Next.



- e From the Standard APIs list, select the following record depending on your QAD ERP application version and then click Done.
 - For QAD EE:

| QdocName | XML Syntax | Version | Route | Procedure | Event |
|----------------------|------------|---------|----------------|-----------|-------------------|
| maintainCustomerData | Qdoc 1.0 | ERP3_1 | UI API Adapter | adcsmt.p | adcsmt-ERP3_1.xml |

- For QAD SE and earlier versions:

| QdocName | XML Syntax | Version | Route | Procedure | Event |
|-------------------------|------------|---------|--------|-----------|------------------|
| deleteCustomerShipTo | QDoc 1.0 | eB_1 | UI API | adcsmt.p | adstmtdeB_1.xml |
| deleteCustomerShipTo | QDoc 1.1 | eB_2 | UI API | adcsmt.p | adstmtdeB_2.xml |
| maintainCustomer | QDoc 1.0 | eB2_1 | UI API | adcsmt.p | adstmtdeB2_1.xml |
| maintainCustomer | QDoc 1.1 | eB2_2 | UI API | adcsmt.p | adstmtdeB2_2.xml |
| maintainCustomerShip To | QDoc 1.0 | eB2_1 | UI API | adcsmt.p | adstmtdeB2_1.xml |
| maintainCustomerShip To | QDoc 1.1 | eB2_2 | UI API | adcsmt.p | adstmtdeB2_2.xml |

f A summary page is displayed. The receiver has been successfully added.

4 Go to Connection Pool Manager and create a UIAPI connection pool for the receiver created.

a In Connection Pool Manager, click Add UIAPI Pool under Add Connection Pool.

b Enter the appropriate configuration settings and click Save.

Note If QAD CRM is installed, you add a database connection to the CRM database in the UIAPI connection setting.

The screenshot shows the 'Configuration Settings Update' dialog box in the QXTend Inbound application. The 'Connections' tab is active. The configuration parameters are as follows:

- Pool Name: QADCSS
- Host: Server08
- Port: 23
- Server Startup Script: login:|netui|Password:|abooolo
- Server Startup Password: *****
- Minimum Connections: 1
- Maximum Connections: 2
- Maximum Failures: 15
- Connections Monitor Frequency: 60000
- Wait time for Idle Connection: 20000
- Max Licensed Agent Retry: 5
- Wait time for Licensed Agent: 20000
- Connection Timeout: 1800000
- Processing Timeout: 600000
- Message Timeout: 10000
- Processing Message Timeout: 10000
- Initializing Timeout: 180000
- Stop On Pause: false
- Operating System Win32/NT: false
- Progress Controller Program: mfw01b.p
- NT Delay: 500
- Connection Setup User ID: userid
- Connection Setup Password: *****
- Domain (If Applicable):

Buttons for 'Save' and 'Cancel' are visible at the bottom of the dialog.

Pool Name. The pool name must match the receiver name for QXI API pools. The pool name displays in the view and delete connection pool menus.

Host. Enter the machine name or IP address of the telnet server.

Port. Enter the port number for the telnet server.

Server Startup Script. Enter the startup script for the telnet session. Specify the telnet server log-in prompts and the responses to these prompts separated with the pipe symbol (|). The standard order is: loginPrompt|userid|passwordPrompt|\$PASSWORD|osPrompt| startScript. For example:

```
log in:|QXtend|Password:|password|$|exec ./qma.QXprod.
```

For Oracle implementations, modify the qma script.

Server Startup Password. Specify the password for the telnet session startup script (maximum 20 characters). The password is encrypted on entry. The startup script substitutes the encrypted password for the \$PASSWORD reference.

Minimum Connections. Enter the minimum number of open connections that the Connection Pool Manager maintains. During startup, the Connection Pool Manager opens this number of connections. As connections are used, it continues to open more so that this number of open connections is maintained, until it reaches the value specified for Maximum Connections.

In general, keep this number as low as effectively possible; for example, 3 on faster systems. On slower systems, increase the number to reduce startup time on new requests.

Maximum Connections. Enter the maximum number of open connections that the system allows. The Connection Pool Manager does not open any more connections than the number you enter.

Important On Windows systems, set this field to 2 or more to ensure successful connections.

Maximum Failures. Enter how many times the Connection Pool Manager attempts to restart an unsuccessful connection. This number is reset when a successful connection is made. You can also reset it by using the Reset Failed Init Count command on the Connection Pool Functions menu.

Connections Monitor Frequency. Enter, in milliseconds, the interval for checking all connections. The default value is 180000 (3 minutes). The interval monitors all connections in all states and closes the connections that have timed out.

Wait Time for Idle Connection. When a connection is requested from the Connection Pool Manager, this timeout specifies the maximum wait for the connection. It is possible that the maximum number of connections has been reached, or new connections are in the initializing state. The default value is 20000 (20 seconds).

Max Licensed Agent Retry. Specify the number of times the system attempts to reserve a licensed agent before returning an exception.

Wait Time for Licensed Agent. Specify the number of milliseconds that the system waits for a licensed agent.

Connection Timeout. Enter, in milliseconds, how long an HTML session can remain inactive before the Connection Pool Manager closes it. The default value is 1800000 (30 minutes).

Processing Timeout. Enter, in milliseconds, how long a connection can be in processing mode. Processing mode indicates a locked or busy screen. The default value is 3600000 (60 minutes). Connection Pool Manager closes locked or busy connections that exceed the default value.

Message Timeout. Enter the interval, in milliseconds, for Connection Pool Manager to wait for a general messaging reply from the telnet server. The default value is 10000 (10 seconds).

Processing Message Timeout. Enter the interval, in milliseconds, for Connection Pool Manager to wait for reply from the telnet server when a connection is in processing mode. The default value is 6666 (6.6 seconds).

Initializing Timeout. Enter the interval, in milliseconds, for Connection Pool Manager to wait for a telnet session to successfully initialize. The default value is 180000 (3 minutes).

Stop on Pause. For QXI, always set the default value to false. This step prevents a transaction from failing when a “Press Spacebar” message is displayed in the target QAD ERP session.

Operating System Win32/NT. Set the value to true if the Progress telnet sessions are executing on a computer with a Windows operating system. Otherwise, set the value to false.

Progress Controller Program. Enter `mfw01b.p` for UI API pools.

NT Delay. You can safely ignore it for QXI connection pools.

Connection Setup User ID. This Connection Setup User ID and the next two entries are the parameters required to connect to the target QAD ERP instance. Enter the valid QAD ERP user ID, such as `qxtend`.

Connection Setup Password. Enter the password for the QAD ERP user ID (maximum 20 characters). The password is encrypted on entry.

Domain. Enter the valid QAD ERP domain if the target instance has domains implemented. Domains were introduced in QAD ERP version 2.1.

Completing Installation Setup

This section describes other tasks required to complete the installation and configuration of QAD CSS. Use these instructions to:

- Set up directory paths to support a number of administrative functions.
- Set up system and order control maintenance features.

Set QAD CSS Directory Paths

Before you can access the QAD CSS database, verify your system variables. Later, these variables can be changed from administrative functions on the QAD CSS menu.

Use MFG/UTIL to set up paths for several images, scripts, and styles directories.

Note If you did not exit from Progress Explorer in step 12 on page 29, do so now; otherwise, the connect fails.

- 1 Start MFG/UTIL and choose Progress Data Dictionary from the Database menu to connect to your QAD CSS database.
- 2 In MFG/UTIL, choose Configure CSS|Set Directory Paths.

Fig. 2.28
Set Directory Paths Screen

- 3 Accept the defaults or enter corrected values for the CSS paths.

Image Path. The virtual directory on the Web server where the QAD CSS images reside. For example, this directory can be defined as:

- An absolute URL, such as `http://www.server.com/qadcsc/images/`
- A relative URL, such as `/qadcsc/images/`

Script Path. The virtual directory on the Web server where the QAD CSS JavaScript files reside. For example, this directory can be defined as:

- An absolute URL, such as `http://www.server.com/qadcsc/scripts/`
- A relative URL, such as `/qadcsc/scripts/`

Style Path. The virtual directory on the Web server where the QAD CSS HTML style sheets reside. For example, this directory can be defined as:

- An absolute URL, such as `http://www.server.com/qadcsc/styles/`
- A relative URL, such as `/qadcsc/styles/`

Temporary Directory. The directory identified in your `qadcsc.pf` file by the `-T` parameter.

MFG/PRO Source Code Directory. The location of QAD ERP compiled code.

QADCSS DB Name. The QAD CSS database name. No extension is required.

Choose OK to save any changes and write the directories to WebSpeed.

Set the Default Data Source

When implementing QAD CSS with QAD ERP, define a default data source to allow the initial system user to log in to the system.

- 1 If you exited MFG/UTIL in the previous step or are not already connected to your QAD CSS database, start MFG/UTIL and choose Progress Data Dictionary from the Database menu to connect.
- 2 In MFG/UTIL, choose Assign Default Data Source|Domain from the Configure CSS menu.

Fig. 2.29

Defining the Default Data Source

The screenshot shows a dialog box titled "CSS UPGRADE eb21". It contains three text input fields. The first field is labeled "Default Data Source:" and contains the text "qadprod". The second field is labeled "Default Domain:" and also contains "qadprod". The third field is labeled "QADCSS DB Name:" and contains "qadcass". At the bottom of the dialog, there are two buttons: "OK" on the left and "Cancel" on the right.

- 3 Enter a valid domain from your QAD ERP installation and enter an equivalent data source for QAD CSS.
- 4 Choose OK to save the changes.

Configure Settings in QAD CSS

Use this section to configure a number of settings within the QAD CSS application.

- 1 Restart the WebSpeed agents.
- 2 Using a Web browser, enter the following URL to connect to the log-in page of QAD CSS:
`http://webServer/webServerScriptsDirectory/wspd_cgi.sh/WService=
brokerName/lg/lg_login.html`
Note In Windows environments, `cgiiip.exe` is typically specified rather than `wspd_cgi.sh`.
- 3 To log in, specify the default user ID and password, `demo` and `demoex`, respectively. At login, you are prompted to change the password for user `demo`.

Update System Control Settings

- 1 From the menu, select Administration|System Control|System Control.
- 2 Review the options and determine which options you want to change to optimally configure the QAD CSS application. Initially, you can accept all defaults. However, note the following fields:
 - Temp Directory reflects the appropriate path to the `/temp` directory used by the application.

- Always select Extend to ERP to enable communication with the QAD ERP database.
- Enter the correct QAD CSS Receiver name and URL for communicating data to QAD ERP through QXtend Inbound. For example, if the QAD CSS Receiver Name is `css_rev`, then QAD CSS Receiver URL is `myQdocWebService` configured in QXtend, such as `http://167.3.129.36:8080/qxi1.6/services/QdocWebService`.
For information about configuring QXtend Inbound, see *QAD QXtend User Guide*.
- Set the Help Path to the virtual directory on the web server where the QAD CSS Help files reside; for example: `/qadcsc/Help/`.

Note After you change the control settings, restart the system for the changes to take effect.

Update Registry Settings

Use System Registry Maintenance to specify a user ID and password that let QAD CSS access QAD ERP.

- 1 From the menu, select Administration|System Registry Maintenance.
- 2 Update these registry values used when logging in to QAD ERP:
 - QADUser, by default `mfg`
 - QADUPasswd, by default blank
 - QADUDomain, by default blank
 - a Enter QADUser in the Search for field and click Search.
 - b Click the QADUser key to display the Detail Registry Editor.
 - c By default, this field is set to `mfg`. If `mfg` is not a valid user in the QAD ERP system you are accessing, change it now.
 - d Repeat these steps for QADUPasswd to secure this user with a password.

Important Make sure that you set up the same user ID with the same password in QAD ERP User Maintenance (36.3.1). Make sure that you have a process in place to update the password for this user within the expiration date, if this security feature is being enforced. If the password prompt displays during login, the integration to fail.

- e Repeat for QADUDomain, entering the default data source setup in “Set the Default Data Source” on page 47.

The system registry also includes numerous other settings that control how you use QAD CSS.

- 3 Update the value for the directory where JavaScript messages are stored.
 - a Enter `js` in the Search for field and click Search.
 - b Click the `jsmsgDir` key to display the Detail Registry Editor.
 - c Change the Key Value to `CSSInstallDir/qadcsc/scripts`.

Note In a multi-tier installation, when the Web server is a Windows server and the application server is on UNIX or Linux, specify the `scripts` directory on the Web server.

 - d Click Save, then click Home.

- 4 Update the following registry values based on the receiver setup in “Setting Up QXtend Inbound”:

QDocNameCustomer. QDoc name for customer maintenance; for example, maintainCustomerData for QAD EE and maintainCustomer for QAD SE and earlier versions.

QDocVerCustomer. Qdoc version for customer maintenance; for example, ERP3_1 for QAD EE and eB2_1 for QAD SE and earlier versions.

Generate Messages

- 1 From the menu, choose Administration Menu|Messages Menu|Error Message.
- 2 Click the Generate button on the left side. The process compiles a JavaScript file containing messages for the system.

Note During use of the application, click the Generate button on the left side whenever you add a message of type JS or support for a new language.
- 3 Click Home to return to the main menu.

Register Your License Key

- 1 Choose Administration Menu|System Control Menu|QAD CSS Clearance Code.
- 2 Enter the customer name as it appears on the license page included in your product package.
- 3 Enter the license key as it appears on the license page.
- 4 Click submit. You are redirected to a secure server URL for registering your product. Depending on how your browser is configured, you may see several messages about security.
- 5 Click Yes to accept the terms of the license agreement.
- 6 The QAD CSS Clearance Code screen redisplay with information about your license, including the number of WebSpeed agents and the QAD CSS modules. Click Update to accept this information.
- 7 Leave QAD CSS running and return to the menu in the next procedure.

Update QAD ERP for Credit Card Processing

If you plan to implement credit card processing, update your QAD ERP installation with several programs and data files. The required files are included on the QAD CSS installation media in QAD ERP version and service pack-specific directories. For example, if you are using MFG/PRO eB2 SP9, use the files in *CSSInstallDir/MFGPRO/eB2/SP9*.

For complete details on implementing credit card processing, see the chapter on B2C order processing in *QAD Customer Self Service Implementation Guide*.

To implement these changes, follow the steps to compile programs and load data files; then update the VeriSign initialization file.

Compile Programs and Load Data

- 1 Locate the files for your QAD ERP release and service pack on the CD.
- 2 Copy the files to your `QADERPInstallDir/xrc`.
- 3 If you are using QAD SE and earlier versions of QAD ERP, load updated schema files.
 - a Start MFG/UTIL and choose Database|Load Database Schema (.df) File.
 - b Connect to the qaddb database and choose OK. Locate the schema file `delta_socc.df` you copied from the QAD CSS installation media.
 - c When the file load is complete, press Enter to close the window.
- 4 Compile the updated program files:
 - a Create a text file named `csscomp.wrk`. Make sure that it contains a listing of the `.p` files you copied.
 - b Start MFG/UTIL and use the `csscomp.wrk` compile list. Set the propath to `QADERPInstallDir/xrc` and the destination directory to `QADERPInstallDir`.

Configure the VeriSign Initialization File

QAD ERP reads settings you define in the VeriSign initialization file (`verisign.ini`) to determine how to handle credit card transactions during invoice post. Locate this file in the QAD ERP PROPATH. Similar settings are defined in QAD CSS in Credit Card Vendor Code Maintenance.

Important Make sure that the settings in the two places are the same or errors occur while processing credit card orders.

Here is a sample `verisign.ini` file.

```
[Verisign]
VendorHost=pilot-payflowpro.paypal.com
VendorPort=443
TimeOut=30
Partner=Paypal
Vendor=QADCSS
User=QADCSS
Password=123
VendorUrl=/transaction
```

Use the following field descriptions to specify values appropriate for your system:

VendorHost. Use `pilot-payflowpro.paypal.com` for the test environment, and `payflowpro.paypal.com` for the production environment.

VendorPort. Specify the port number supplied by Paypal, typically 443. API uses the number when transactions are sent to be authorized.

Partner. Specify Paypal.

Vendor. Paypal user account and password after the equal sign = of the Vendor, User, and Password respectively.

User. Specify the user name that identifies you to your credit card processing company.

Password. Specify the secure password associated with the user previously entered. The password is required for transmitting credit card transactions.

Update the CSS system registry as follows:

| Module | Override | Key | Key Value |
|--------|----------|-----------------------|--|
| cc | No | CreditCard VendorCode | Update the vendor name maintained in Credit Card Vendor Maintenance. |

Modify the existing Credit Card Vendor record by accessing Credit Card Module\Credit Card Vendor Maintenance in CSS and making the following changes:

- Change Partner Name to `Paypal`.
- Change Partner Host's Name to `pilot-payflowlink.paypal.com` for the test environment, or `payflowlink.paypal.com` for the production environment.
- Set Partner API Host's Name to `pilot-payflowpro.paypal.com` for the test environment, or `payflowpro.paypal.com` for the production environment.

Setting Up Integration of QAD CSS and QAD Configurator (Optional)

If you plan to implement the integration of QAD CSS and QAD Configurator (5.3 and later) so there can be customer self service orders for configurable products, follow these steps.

Prerequisite: Make sure that QAD Configurator has already been installed.

- 1 In the `CSSInstallDir/qadcsc/css/cfg` directory, open the `config.i` file and set `configurator-integrated` to Yes.
- 2 In the `CSSInstallDir/qadcsc` directory, open the `qadcsc.ini` file and add `CPDInstallDir`, `CPDInstallDir/LanguageCode`, and `CPDInstallDir/cop_xrc` to the `PROPATH`; also add the database of QAD Configurator in the `qadcsc.ini` file.
- 3 Compile the QAD CSS source code.
- 4 In System Control, set `Integrated with Configurator` to Yes.

Use System Registry Maintenance to specify the value of the registry `ConfiguratorWSURL` as `http://cfgWebSpeedServerHostName/cgi-bin/wspd_cgi.ksh/WService=cfgWebSpeedBrokerName`

Setting Up QAD CSS for Single Sign-on (Optional)

Using single sign-on (SSO), a user can log in once and gain access to all systems, including QAD CSS, without being prompted to log in to each system again.

Before you set up CSS for single sign-on, make sure that you have installed a SAML extension; the extension contains a filter that intercepts requests to CSS login. For more details about SAML extension installing, see “Installing SAML Extension (Optional)” on page 52.

Also make sure that you have installed QRA, which is necessary for single sign-on and user synchronization.

To set up QAD CSS for single sign-on, take the following steps:

- 1 Make sure that the `usersync` directory is in the CSS WebSpeed server `PROPATH`.
- 2 Use WebSpeed Workshop to compile the following files:
 - `lg_login.html`
 - `lg_samllogout.html`
 - `lg_samlunauth.html`
- 3 Open the QAD CSS application as an administrator.
- 4 Use System Registry Maintenance to add the following registry records:

Table 2.6
Registry Records

| Key | Value | Override |
|-------------------|--|----------|
| AppserverURL | AppServerDC://vmejt02.qad.com:16569/QA DQRA_AS-sessionModel Session-free | No |
| AppserverUID | Leave the key value blank. | No |
| AppserverPWD | Leave the key value blank. | No |
| AppserverParam4 | Leave the key value blank. | No |
| ServiceContextUID | mfg | No |
| ServiceContextPWD | Leave the key value blank. | No |
| SAMLmsg | Leave the key value blank. | No |
| AppserverLoc | Enter the location of the AppServer, such as <code>http://vmm9d01.qad.com:16560/qad-css/</code> | No |

Note It is possible that your values are different from the values listed in the table.

Installing SAML Extension (Optional)

Security Assertion Markup Language (SAML) extension is an XML-based extension for exchanging authentication and authorization data. Before you set up CSS for single sign-on, make sure that you have installed a SAML extension.

For browser selection, it is recommended that you use the Firefox browser because you can use the Firefox SAML Tracer to view SAML messages sent through Firefox.

Note The installation steps include a number of manual steps. Problems with the installation are often caused by typos. For example, you accidentally add a backslash or a blank character at the end of a URL, which can cause the installation to fail. So when entering URLs, double-check your work to make sure that the URLs are correct.

- 1 Deploy the `qad-css.war` file into Tomcat using the Tomcat manager.

Note Although you have deployed the Web application into Tomcat, it does not start unless you complete the SAML configuration. Spring Security SAML Extension version 1.0.0 has introduced some code changes that are required to be accounted for in the Security Context XML configuration.

2 Install WebSpeed CGI Messenger.

- a** Copy `$DLC/bin/cgiip` to the `WEB-INF/cgi` directory.
- b** Copy `$DLC/bin/wspd CGI.sh` to the `WEB-INF/cgi` directory.
- c** Make sure that the files are executable.
- d** Modify the `wspd CGI.sh` file so it is configured properly for your deployment. You can enable option 4 for the QADCSS_WS WebSpeed service and disable the other options. Double-check the `$DLC` setting.

```
# option 1 using host_name and port_num
# $DLC/bin/cgiip pegasus 5001
# option 2 using a service name defined in $DLC/ubroker.properties
# $DLC/bin/cgiip -i wsbroker1
# option 3 the "defaultService" defined in $DLC/ubroker.properties
# $DLC/bin/cgiip
# option 4 using a specific properties file name
$DLC/bin/cgiip -i QADCSS_WS -f $DLC/properties/ubroker.properties (you should input your
service name and ubroker file path in your environment)
# option 5 using a specific properties file name with the "defaultService"
# $DLC/bin/cgiip -f ./mybroker.properties
```

Note You can only enable one option. Use the `#` character to comment out unused options.

3 Generate Client ID.

- a** Open Client ID Maintenance.

Note This function is only available in QRA-enabled environments.

- b** Generate a new Client ID.
- c** Provide a description and set Active to Yes.
- d** Copy the Client ID and Client Secret.

4 Update the SAML Configuration.

- a** A default SAML configuration file is included: `WEB-INF/classes/security/securityContext.xml.default`. To use it, remove the `.default` extension.
- b** Open the `WEB-INF/classes/security/securityContext.xml` file.
- c** Update the `entityId` value with the URL of your web application.

5 Install the Identity Provider (IdP) metadata.

- a** Your Identity Provider provides you with a metadata file. Create the `idp.xml` file.
- b** Copy your IdP metadata to the `WEB-INF/classes/security/idp.xml` file. The metadata file contains the entityID of the Identity Provider, the X509 Certificate, the name ID format and the supported SSO bindings.

6 Generate the Service Provider (SP) metadata.

- a** Restart your web application using the Tomcat manager.
- b** Open a web browser to the Service Provider metadata URL of your web application. The SP metadata is then downloaded to your computer.

Note If your Identity Provider supports metadata, you can import this file directly. Otherwise, extract information from the metadata file, including *entityID*, *AssertionConsumerService*, and *SingleLogoutService*.

Your Identity Provider requires you to provide some information, which is listed in the following table.

Table 2.7 Information Required by the Identity Provider

| | |
|------------------------|---|
| SAML Consumer URL | <code>\${AssertionConsumerService}</code> |
| SAML Single Logout URL | <code>\${SingleLogoutService}</code> |
| SAML Audience | <code>\${entityID}</code> |
| SAML Recipient | <code>\${AssertionConsumerService}</code> |

7 Add key and value in CSS System Registry Maintenance.

8 Update JSP files.

Update the `index.jsp` and `logout.jsp` files in the CSS Web application root directory to reference the correct WebSpeed broker for CSS. Replace `QADCSS_WS` with the name of your CSS WebSpeed broker.

9 Log in to CSS.

You can log in to CSS by URL. The Web page redirects you to the CSS login page.

Note Because you log in to the OneLogin with your administrator account, add or synchronize this account to the CSS database first. Also, compile `lg_samlunauth.html` and `lg_samllogout.html` manually and use WebSpeed Workshop for error pages when logging in.

Setting Up QAD CSS for User Synchronization (Optional)

To set up QAD CSS for user synchronization, take the following steps:

- 1 Open the `/css/usersync/UserDetails.xml` file in your editor, and fill in the details of users to be added. If the details are unknown, or there are large quantities of users, then the process uses the default values in `SyncConfig.xml`.
- 2 Open the `/css/usersync/SyncConfig.xml` file in your editor, and edit values for the following SyncData elements:

| Field | Value |
|-----------------|--|
| AppserverURL | Enter the AppServer URL. |
| AddUsers | Enter True or False. |
| ModifyUsers | Enter True or False. |
| InactivateUsers | Enter True or False. |
| ConfigDir | Enter the location for the <code>css/cfg</code> directory. |

Note Defaults are defined for Company Association, Role, User Type, and Data Source. These defaults are dependent on the installation site. Currently, the value for *defaultPassword* is the encryption value for `temppass`.

- 3 Open the `/css/usersync/DSMLConfig.xml` file in your editor, and edit values for the following DSMLData elements:

| Field | Value |
|-------------|--|
| ttAction | Enter <code>Update</code> for LDAP(s), which is used for updating and adding. Enter <code>Inactivate</code> for LDAP(s), which is used for inactivation. |
| ttTimeCheck | Enter <code>True</code> or <code>False</code> . The value to use depends on whether <code>timestamp</code> is used for part of LDAP Filter. |
| ttURL | Enter DSML Serverlet URL. |
| ttUID | Change <code>***</code> to an LDAP-authorized user ID. |
| ttPWD | Change <code>*****</code> to an LDAP user ID password. Use an encrypted password. The encryption can be found by running the script <code>Encrypt.sh</code> . The value to be encrypted is passed as a parameter. The output is the encryption. The process uses the same hard-coded encryption key as the synchronization process. |
| ttFilter | <code>(cn=*)</code> represents the list of users to be added or modified. A wild-card character <code>*</code> refers to all. <code>(cn=m9d)</code> represents a user whose user ID is <code>m9d</code> . |

Note There are multiple DSMLData entries for the action `Update`. You can have as many entries as there are LDAP repositories. This situation also applies for the DSMLData entries for the action `Inactivate`.

- 4 Open the file `/css/usersync/SyncUsers.sh` in your editor, and change the following settings:
 - `DLC`
 - `PROPATH`
 - `-pf` parameter, which refers to your database connection property file
- 5 Complete the installation of User Synchronization. To execute the `Sync`, navigate to `/css/usersync` in Putty and run the `./SyncUsers.sh` file.

Note The file `encrypt_devel.i` should not be included. This file is an unencrypted version of `encrypt.i` and should not be included for installation.

Upgrading QAD CSS Using MFG/UTIL

This chapter describes how to upgrade from previous QAD CSS releases to the latest CSS version using MFG/UTIL on UNIX or Windows platforms.

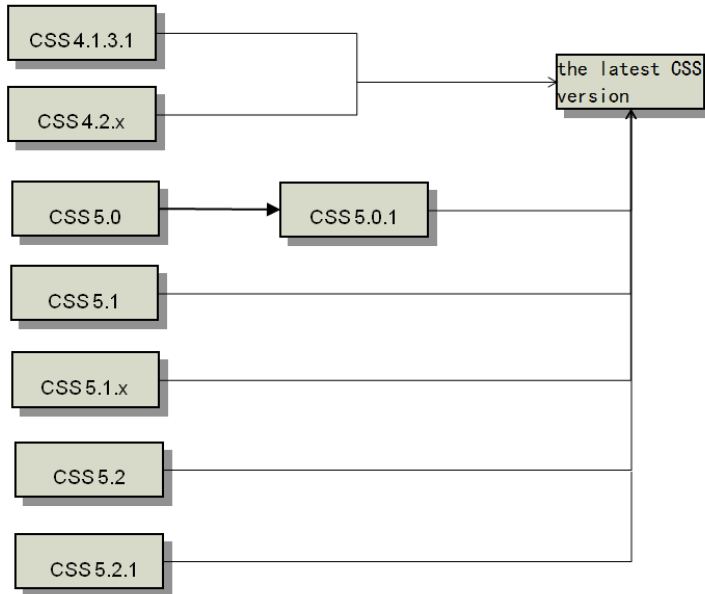
| | |
|--|-----------|
| Overview | 58 |
| Installing New QAD CSS Media | 59 |
| Setting Up QAD CSS Environment Values | 59 |
| Converting the Database | 61 |
| Starting the Database Server | 64 |
| Modifying the QAD CSS Setup | 64 |
| Applying Patches | 64 |
| Verifying QAD CSS Setup | 64 |
| Generating a WebSpeed Error File | 64 |
| Compiling QAD CSS Source Code | 64 |
| Completing Conversion Setup | 64 |
| Final Steps | 64 |

Overview

This chapter describes how to upgrade from previous QAD CSS releases to the latest CSS version using MFG/UTIL on UNIX or Windows platforms. If you are going to manage CSS patch installation with YAB, see “Installing or Upgrading QAD CSS Using YAB” on page 65.

Figure 3.1 illustrates the conversion paths of different CSS versions.

Fig. 3.1
QAD CSS Conversion Paths



Except for CSS 5.0, which is upgraded to CSS 5.0.1 first, all earlier versions of CSS can be directly upgraded to the latest CSS version using the following general steps:

- Upgrading Progress to OpenEdge 10.1A or Above
- Installing New QAD CSS Media
- Setting Up QAD CSS Environment Values
- Generating Database Scripts
- Converting the Database
- Starting the Database Server
- Modifying the QAD CSS Setup
- Applying Patches
- Verifying QAD CSS Setup
- Generating a WebSpeed Error File
- Compiling QAD CSS Source Code
- Completing Conversion Setup
- Final Steps

For information on upgrading from CSS 5.0 to CSS 5.0.1, see *QAD Customer Self Service Installation Guide version 5.0.1*.

Upgrading CSS

Upgrading Progress to OpenEdge 10.1A or Above

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

Installing New QAD CSS Media

These instructions assume that you are installing to a new location and not overwriting your existing QAD CSS system. After upgrading and testing the new QAD CSS version, you can switch over to your production system.

Follow the same steps in “Running the Installation Script” on page 11.

Setting Up QAD CSS Environment Values

Before beginning the conversion, make sure to define some environment variables required by MFG/UTIL.

- 1 Open the `mfgutil` script file under `CSSInstallDir` using a text editor and append the `-cpinternal` and `-cpstream` startup parameters to the last line.

For example, if the last line is:

```
$DLC/bin/_progres -p /home/envs/dlc101c/xmfgusrc/mfgutil.p -c 500 -s 63 -D 50 -TM 31 -TB
31 -B 1000 -y y 1920 -d mfy -ininame mfguprog.ini
```

Append the parameters and the last line becomes:

```
$DLC/bin/_progres -p /home/envs/dlc101c/xmfgusrc/mfgutil.p -c 500 -s 63 -D 50 -TM 31 -TB
31 -B 1000 -y y 1920 -d mfy -ininame mfguprog.ini -cpstream codepage -cpinternal codepage
```

Specify the same code page of the target QAD CSS database. For example, if your QAD CSS database encoding is utf-8, then specify the following:

```
-cpinternal utf-8, -cpstream utf-8
```

- 2 Launch MFG/UTIL from the `CSSInstallDir` using the following command:

```
./mfgutil
```

For Windows, launch MFG/UTIL from the icon on the Start menu.

- 3 Select CSS Setup from the Configure CSS menu. Use the following screen and values to update your QAD CSS configuration.

Fig. 3.2
CSS Setup Screen

Table 3.1
CSS Setup Fields

| Field | Description |
|----------------------------|---|
| CSS Database Directory | It is <i>CSSInstallDir/database</i> by default. |
| CSS Install Directory | The <i>CSSInstallDir</i> defined during install. |
| Progress Install Directory | The Progress install directory as identified during the install. |
| MFGPRO Install Directory | The <i>QADERPInstallDir</i> for the QAD ERP instance CSS connects with. |
| WebServer Host Name | The server name or IP address of the Web server. The name also includes a domain name. If the domain name is not included, the data entered in the screen is not saved properly. The <i>.qad.com</i> in the sample screen is the domain name. |
| WebSpeed Broker Name | The name of the broker you plan to use; this guide uses <i>qadcss</i> as an example. |
| WebSpeed Broker Port | An available port specified as a part of the “Preparatory Information” on page 8. |

- 4 Choose OK to save the changes.

Generating Database Scripts

To convert the database, generate scripts for starting and stopping the database first.

Follow the same steps in “Generate Database Scripts” on page 17.

Converting the Database

The database conversion is completed using a set of options on the Upgrade CSS menu in MFG/UTIL.

Conversion Overview

Completing the conversion involves the following activities, represented by options on the MFG/UTIL Upgrade CSS menu:

- 1 Copy your existing QAD CSS database to the latest CSS version installation directory.
- 2 If you upgrade your Progress version to OpenEdge 10.1A, first convert your CSS database to OpenEdge release 10.1A.
- 3 If the code page of your old QAD CSS database is different from the one you want to implement for your new database, edit file `mfgutil` to change the following parameters:

```
-cpstream NewCodePage -cpinternal NewCodePage
```
- 4 Load the delta schema file containing definitions of new QAD CSS tables and existing table modifications.
- 5 Run a conversion script that dumps the data from your existing database and updates it for schema changes introduced in the latest CSS version.
- 6 Load the data that was dumped in the previous step into the updated tables.
- 7 For QAD EE, convert the CSS database's code page to utf-8.
- 8 Update the version to reflect the latest CSS version.
- 9 Clean the credit card information in QAD CSS database and the `qaddb` database.
- 10 For QAD CSS versions before 4.2.1, enhanced security for program pages and associated child pages was introduced in the 4.2.1 release. To implement this feature, run another option that builds the security information based on the menu structure in your database.
- 11 Specify a default domain for initial login.

Note The option Dump Data from Database on the Upgrade CSS menu is currently not required since the data is dumped automatically during the first step.

Running the Conversion

Use the following procedures to convert your existing QAD CSS database.

Note If you are upgrading from version 5.2.1, skip the steps of loading the delta schema file.

Fig. 3.3
Converting Database

File Database Programs Scripts Configure CSS **Upgrade CSS** Help

```

Load Database Schema (.df) File
Dump Data From Database
Load Data into Database
Build Security Tree
Update Product and Version
Assign Default DataSource/Domain
Convert CSS
Convert CSS To UTF8
Clean Credit Card Info

```

- 1 Copy the database to be upgraded from its existing location to the `/database` directory under the latest CSS version installation directory.

- a Change your current directory to `CSSInstallDir`.
- b Use `procopy` to copy your previous version database to:

```
CSSInstallDir/database/qadcscs
```

- 2 If you upgraded your Progress version to OpenEdge 10.1A, convert the CSS database to OpenEdge 10.1A. For information on converting previous versions of Progress databases to OpenEdge release 10.1A, see *OpenEdge Data Management Database Administration Guide*.

- 3 If the code page of your old QAD CSS database is different from the one you want to implement for your new database, edit the `mfgutil` file to change the following parameters:

```
-cpstream NewCodePage -cpinternal NewCodePage
```

- 4 (Skip this step when you upgrade from version 5.4 or later.) Choose Load Database Schema (.df) File from the Upgrade CSS menu. In the Connect Database screen, connect to your QAD CSS database.

- a In the Load Data Definition screen, use the Browse button to locate the appropriate .df file in the subdirectory with the name of the QAD CSS release you are upgrading from:

```
CSSInstallDir/defs/upgrade/SourceCSSVersion
```

The file name is also based on your source QAD CSS version; for example, when you are upgrading from QAD CSS 4.2.1, choose:

```
CSSInstallDir/defs/upgrade/4.2.1/delta421_523.df
```

- b When you have selected the correct file, click OK. The system loads the new schema into the database.

Note For CSS 4.2.2, make sure to load two files—`/upgrade/4.2.2/delta422_423.df` and `/upgrade/4.2.3/delta423_523.df`.

- c When the load is complete, choose Close to continue.

- 5 Start MFG/UTIL and choose Convert CSS from the Upgrade CSS menu.
 - a In the CSS Upgrade Original Version window, enter the QAD CSS version number you are upgrading from; for example, 5.1.1.
 - b In the Connect Database screen, connect to the target CSS database. The log window displays showing a successful connection.
 - c The system performs CSS data conversion and dumps all the .d files in the *CSSInstallDir/convdata* directory. When the process is complete, choose Close to continue.
- 6 Choose Load Data into Database from the Upgrade CSS menu.
 - a In the Connect Database screen, connect to your CSS database.
 - b In the Load Data Contents screen, use the Browse button to locate the *CSSInstallDir/convdata* directory and choose OK.
 - c The system loads updated data into the database. When the loading is complete, choose Close to continue.
- 7 For QAD EE, convert the CSS database's code page to utf-8. Choose Convert CSS To UTF8 from the Upgrade CSS menu and connect to the CSS database you want to convert. In the Load Data Definition screen, choose *\$DLC/prolang/utf/ICU-UCA.df*.
- 8 Choose Update Product and Version from the Upgrade CSS menu.
 - a Enter the correct version name and database name.
For the current version, enter *ex 5.4.1* as the version name.
The database is the QAD CSS database name located in the *CSSInstallDir/database* directory.
 - b The Log window appears and shows progress. Choose Close to continue.

Note When updating licensing records, it is possible that the screen displays errors regarding duplicate records. You can safely ignore these records.
- 9 Choose Clean Credit Card Info from the Upgrade CSS menu.
 - a In the Connect Database Screen, locate */dr01/qadapps/css/database/qadcass.db* to connect to your QAD CSS database, and choose OK.
 - b Connect to the *qaddb* database and choose OK.
- 10 If you are upgrading from QAD CSS 4.2.1, skip to step 11.
Upon completion, choose Build Security Tree from the Upgrade CSS menu.
 - a In the Connect Database screen, connect to your QAD CSS database.
 - b The system builds a security tree for all the CSS modules. When the process is complete, choose Close.
- 11 Choose File|View Log to view the MFG/UTIL log file and ensure that the conversion completed successfully.

Starting the Database Server

Follow the same steps in “Starting the Database Server” on page 21.

Modifying the QAD CSS Setup

If you also upgrade your Progress or WebSpeed version as part of your QAD CSS migration, tailor the configuration of these products for QAD CSS. Follow the instructions in “Installing QAD CSS Using MFG/UTIL” under:

- “Setting Up WebSpeed” on page 24
- “Configuring QAD CSS” on page 31

For more information on using the sample WebSpeed scripts supplied with QAD CSS, see “Administering WebSpeed” on page 39.

Applying Patches

Follow the same steps in “Applying Patches” on page 36.

Verifying QAD CSS Setup

Follow the same steps in “Verifying QAD CSS Setup” on page 37.

Generating a WebSpeed Error File

Follow the same steps in “Generating a WebSpeed Error File” on page 40.

Compiling QAD CSS Source Code

Follow the same steps in “Compiling QAD CSS Source Code” on page 41.

Completing Conversion Setup

Follow the same steps in “Completing Installation Setup” on page 45.

Final Steps

If you want to integrate QAD CSS with QAD Configurator, follow the steps in “Setting Up Integration of QAD CSS and QAD Configurator (Optional)” on page 51.

Installing or Upgrading QAD CSS Using YAB

This chapter describes how to install or upgrade QAD CSS using YAB.

Overview 66

Installing or Upgrading QAD CSS Using YAB 66

(Optional) Modifying the YAB Configuration File 67

Overview

QAD CSS is an add-on product that can be installed into QAD Enterprise Edition. Starting with 2016 EE, you can use YAB to install and upgrade QAD CSS. YAB is a deployment and management toolset that covers all products installed into an Enterprise Edition environment.

Note QAD CSS versions prior to 5.4.2 are not supported by YAB.

For instructions of installing and upgrading QAD CSS on QAD EA with MFG/UTIL, see “Installing QAD CSS Using MFG/UTIL” on page 7 and “Upgrading QAD CSS Using MFG/UTIL” on page 57.

After installation or update, the system requires some configuration to make it ready for use. For details, see “Post-installation Tasks” on page 71.

Installing or Upgrading QAD CSS Using YAB

Follow these steps to install or upgrade QAD CSS using YAB:

- 1 Download the latest YAB version from QAD Release Fulfillment and move it to the directory where QAD Enterprise Edition is installed.
- 2 Install the latest YAB version by running the following command:

```
yab install <yab patch>
```
- 3 Download the CSS package from QAD Release Fulfillment. Place the zip file in a convenient location; for example, the `/qad/install` directory on the server where CSS will be installed.
- 4 Navigate to the directory where Enterprise Edition is installed.
- 5 (Optional) Modify your YAB configuration with specific settings for your installation. See “(Optional) Modifying the YAB Configuration File” on page 67.

- 6 Enter:

```
yab install <zip file>
```

For example:

```
yab install css-src-yab-compatible-5.4.3.0.zip
```

```
yab install css-xrc-yab-compatible-5.4.3.0.zip
```

- 7 The system performs additional CSS installation steps.

Note If the installation fails, check the `./build/logs/yab.log` file.

- 8 Enter the license code for your product.

The installation creates the following directory in the local catalog:

```
./build/catalog/packages/css
```

This directory contains the CSS source code, as well as `dmp` standard files. The CSS compiled code is stored under `./dist/css`.

(Optional) Modifying the YAB Configuration File

This section contains the YAB information specific to the CSS product.

The only configuration file that you should modify is the `configuration.properties` file, which is used to record changes to the standard configuration defaulted by the system.

To apply your configuration, follow these steps:

- 1 Define your configuration settings in the `configuration.properties` file located in the `build/config/` folder.

This file only contains settings that differ from the base product configuration (`build/config/system`) and can initially be empty. To override a setting, add the setting to this file and adjust the value. If the setting is already defined in the file, you can simply adjust the value.

Table 4.1 lists the settings that you can change.

Table 4.1
Configuration Properties

| Key | Default | Usage |
|--|---|---|
| <code>qadcsc.languages</code> | us | This setting determines the system language and should match entries in the <code>encoding.dat</code> file. Currently, the system supports the following languages: Castilian Spanish (cs), Czech (cz), Dutch (du), French (fr), German (ge), Italian (it), Japanese (jp), Korean (ko), Latin Spanish (ls), Polish (pl), Portuguese (po), Traditional Chinese (tw), English (us), Simplified Chinese (ch) By default, the system uses the value <code>us</code> . To have multiple languages on the system, define the value as a string of comma-separated language codes; for example, " <code>us, ch, cs, po</code> ". |
| <code>qadcsc.configurator-integration-enabled</code> | False | This setting determines whether to integrate with QAD Configurator. By default, the feature is disabled. |
| <code>customizations.qadcsc.dir</code> | <code>\${customizations.dir}/qadcsc</code> | This setting determines the directory in which CSS customizations are organized. |
| <code>patches.qadcsc.dir</code> | <code>\${patches.dir}/qadcsc</code> | This setting determines the directory in which CSS patches are organized. |
| <code>ws.qadcsc.name</code> | <code>ws-css</code> | This setting determines the WebSpeed server name. |
| <code>ws.qadcsc.portnumber</code> | inherited from <code>ws._base.portnumber</code> | This setting determines the WebSpeed server port number. |

| Key | Default | Usage |
|---|---|---|
| <code>ws.qadcass.fileUploadDirectory</code> | <code>\${ws.qadcass.work-dir}/upload</code> | This setting determines the directory in which WebSpeed codes are organized. |
| <code>qadcass.receiver.name</code> | QADCSS | This setting determines the QAD CSS Receiver name for communicating data to QAD Enterprise Applications through QXtend Inbound. |

2 Execute the `update` command to apply the configuration change:

```
> yab update
```

You can use the `update` command to apply any change to the system.

The configuration changes, which are limited to updating application configuration files and database settings, can be applied with the quicker `reconfigure` command.

```
> yab reconfigure
```

Note The `reconfigure` command does not restart the environment. It might be necessary to restart the environment or specific servers for the application to recognize the change.

Upgrading from Non-YAB Version to YAB Version

This chapter describes how to upgrade QAD CSS from a non-YAB version to a YAB version.

Overview 70

Upgrade Steps 70

Overview

This chapter covers the instructions for upgrading QAD CSS from a non-YAB version (prior to 5.4.2) to a YAB version. Both MFG/UTIL and YAB tools are used in this upgrade process.

Upgrade Steps

Follow these steps to upgrade QAD CSS from a non-YAB version to a YAB version:

- 1 Run `./install` from QAD CSS media and only copy the first component, which includes QAD CSS database, system data, and toolset files.
- 2 Copy the old QAD CSS database of the non-YAB version to the new QAD CSS database location to replace the new database.
- 3 Do the schema updating and conversion if necessary. (The QADDB database and the QADADM database are not required to be updated or converted.) See “Upgrading QAD CSS Using MFG/UTIL” on page 57.
- 4 Install QAD CSS using YAB. See “Installing or Upgrading QAD CSS Using YAB” on page 65.

- 5 Back up the QAD CSS database that is generated from step 2 to a specified location by running the following command:

```
probkup <non-yab install DB> <backup DB>
```

- 6 Copy the backup database to a location that is accessible from the YAB environment.
- 7 Run the following command to stop the YAB environment:

```
yab stop
```

- 8 Delete the QAD CSS database that is generated from step 4 by running the following commands:

```
rm yabInstallDBLocation/<css db name>*
rm yabInstallDBLocation/extent/<css db name>*
```

- 9 Generate a new YAB-version QAD CSS database by running the following commands:

```
prorest <backup DB> yabInstallDBLocation/<css db name>
```

- 10 Run the following command.

```
yab update -clean
```

Post-installation Tasks

This chapter describes the post-installation tasks so the system is ready for use.

Updating Agent Properties for QAD CSS Broker 72

Updating Agent Properties for QAD CSS Broker

After you have successfully completed the installation and verified the setup, complete the final steps in your production environment.

In Progress Explorer, update agent properties for the QAD CSS broker:

- 1 Click Advanced Features under the Agent section. Set the debug mode to Disabled and click OK.
- 2 Click Pool Range under the Agent section. Set the value based on the number of licensed agents that you have registered and click OK.
- 3 Complete the QAD CSS implementation using *QAD Customer Self Service Implementation Guide*.

Appendix A

Troubleshooting

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

| | |
|----------------|--|
| Issue | Unable to start CSS WebSpeed Server |
| ID | AA-93586 |
| Symptom | After you have installed CSS 5.4.1, you cannot start CSS WebSpeed Server. In the CSS WebSpeed Server log, you get the following error message: Could not find class or interface com.qad.qra.config.QraConfig. (12886) |
| Cause | One program in CSS 5.4.1 references latest QRA component and therefore cannot run in previous QRA versions. |
| Fix | <ul style="list-style-type: none"> • Go to QAD KnowledgeBase, article ID AA-93586, and get the patch <code>lib_sys_superex6.p</code>, which is attached to the article. • Put the patch in the <code>css/lib</code> directory and overwrite the original file. Compile when necessary. |

| | |
|----------------|--|
| Issue | Unable to load CSV file |
| ID | AA-97101 |
| Symptom | After you load an Excel CSV file using the Catalog Load utility, you get a blank screen. |
| Cause | Excel converts CSV files to a format that CSS cannot read. |
| Fix | <ul style="list-style-type: none"> • Open a new spreadsheet in Excel. • Set the format of all cells to Text. • Copy and paste the contents of the CSV file into the new Excel spreadsheet. • Modify the content. • Save the spreadsheet as Text formatted as tab delimited. • Load the new file into CSS using Catalog Load. |

Product Information Resources

QAD offers a number of online resources to help you get more information about using QAD products.

[QAD Forums \(community.qad.com\)](https://community.qad.com)

Ask questions and share information with other members of the user community, including QAD experts.

[QAD Knowledgebase \(knowledgebase.qad.com\)*](https://knowledgebase.qad.com)

Search for answers, tips, or solutions related to any QAD product or topic.

[QAD Document Library \(documentlibrary.qad.com\)](https://documentlibrary.qad.com)

Get browser-based access to user guides, release notes, training guides, and so on; use powerful search features to find the document you want, then read online, or download and print PDF.

[QAD Learning Center \(learning.qad.com\)*](https://learning.qad.com)

Visit QAD's one-stop destination for all courses and training materials.

*Log-in required

Index

A

AdminServer 21
 starting 21
Apache Web server
 virtual directories 33

B

BI file, truncating 17
broker, WebSpeed 25
 logging settings 27
 starting 37

C

CD
 installing 11
 mount commands 11
cgiiip.ex 41
clearance code 49
code
 compiling 41
 compiling QAD CSS code 41
compme3.1.html 41
conversion
 converting database 61
credit card
 installation steps 49, 64, 70
 VeriSign ini file 50
CSS Guided Setup 15

D

database
 creating 12
 loading data 16
 QADCSS DB Name field 46
 start and stop scripts 20
defaultServerGroup 23
deployment options 4
 distributed 4
 single 4
directories
 defining 45

E

Error Message Maintenance 49
Extend to ERP 48

I

Image Path 46
installation
 CD media 11

deployment options 4
requirements 5
script 11

J

JavaScript messages 48
jmsgDir 48

L

logging settings
 broker 27

M

message
 Javascript 48
MFG/UTIL
 CSS Setup Screen 14, 60, 62
 database creation 12
 Generate Sample ubroker.properties File 30
 Setup Directory Paths 45
 Upgrade CSS options 63
mfww01b.p 45
mount commands, UNIX 11

P

password 48
permissions 8
Pool Range 28
Progress Explorer
 setting up Progress servers 24
 setting up remote machines 25
 starting database 21
 WebSpeed setup 24
PROPATH
 verifying 39

Q

QAD CSS
 architecture 3
 Clearance Code 49
 deployment 4
 directory paths 45
QAD Database Builder 15
QAD ERP
 credit card updates 49, 64, 70
 Source Code Directory 46
 User Maintenance 48
QADUDomain 48
QADUPasswd 48
QADUser 48

QXtend Inbound 42

R

registration 49
requirements, installation 5

S

Script Path 46
Serial Number 49
server
 starting 21, 64
services file 20
software prerequisites 67
structure file. 15
Style Path 46
System Control Table Maintenance 47
System Registry Maintenance 48

T

telnet
 server 44
Temporary Directory 46
truncate BI file 17

U

ubroker.properties
 configuring UNIX 30
 configuring Windows 24

 two-tier deployment 35
UI API
 controller program 45
User Maintenance (36.3.18) 48
user, QAD ERP 48
utils_ms 19, 39
utils_ux 19, 39

V

virtual directories 32

W

Web server 6
 virtual directories 32
WebSpeed
 administration 39
 broker setup 25
 Pool Range setting 28
 setting up 24
WebSpeed Messenger
 two-tier deployment 34
WebSpeed Workshop 38
 compiling code 41
wspd_cgi.sh 41

Y

YAB 66