Installation Guide

QAD Customer Self Service (QAD CSS)

Installation Overview
Installing QAD CSS
Upgrading QAD CSS

78-0887-5.2
QAD CSS Version 5.2
March 2012

CSS_IG_v52.pdf/hes/c6s

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

Contents

Change Summary

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

Date/Version	Description	Reference
March 2012/CSS 5.2	Updated the Hardware and Networking Requirements section	Page 5
	Added steps to enable the integration of QAD CSS and QAD Configurator	Page 52
September 2011/CSS 5.1.3	Rebranded for CSS 5.1.3	-
	Added two languages, Czech and Korean, for cpstream settings	Page 14
	Changed `css.pf` to `qadcss.pf`	Page 29
	Added `binaryUploadMaxSize=-1` for the sample file	Page 32
	Added a step for versions earlier than 2011 EE in Configuring QAD CSS	Page 33
	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 50
	Added a step Generating Database Scripts in Upgrading QAD CSS	Page 54
	Changed `51.df` to `513.df`	Page 57
	Updated steps for CSS database conversion	Page 58

Chapter 1

Installation Overview

This chapter includes an overview of the QAD Customer Self Service (QAD CSS) product architecture and various deployment options. It then provides details needed to prepare for installation.

Introduction

QAD Customer Self Service (QAD CSS) is a Web-based storefront application designed to work with QAD's 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.

These 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 system administrator should also be 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 supported by several other technologies:

- 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:

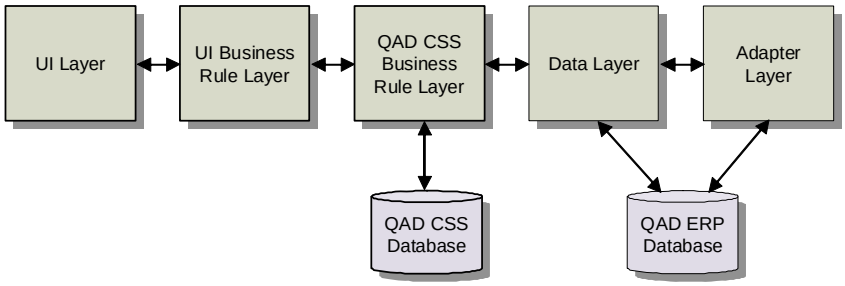
- User interface (UI) layer. This layer presents and collects information that interacts with the user interface. It is written using HTML, Progress SpeedScript, and JavaScript. The UI layer creates the main container for the QAD CSS application and works with the UI business rule layer to dynamically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

  The UI layer ensures that the JavaScript and style sheets are included. In this context, JavaScript validates 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 that include 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, you should 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.

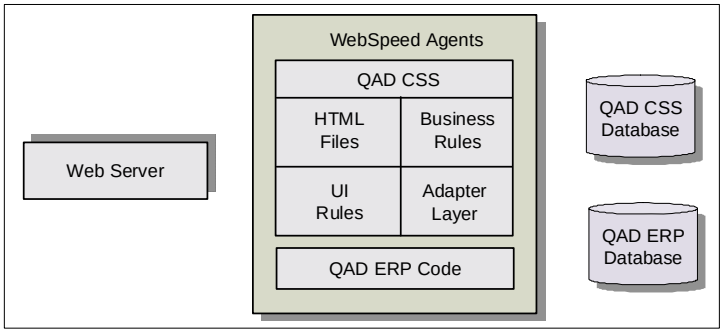
Deployment Examples

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.

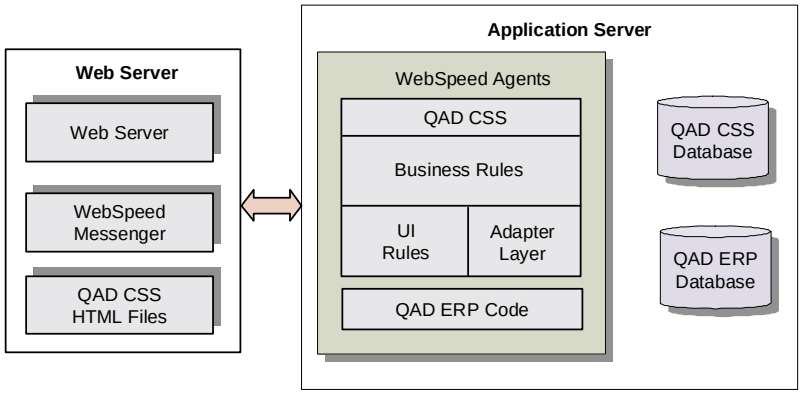
Fig. 1.2
Simple Deployment of QAD CSS



Distributed Deployment

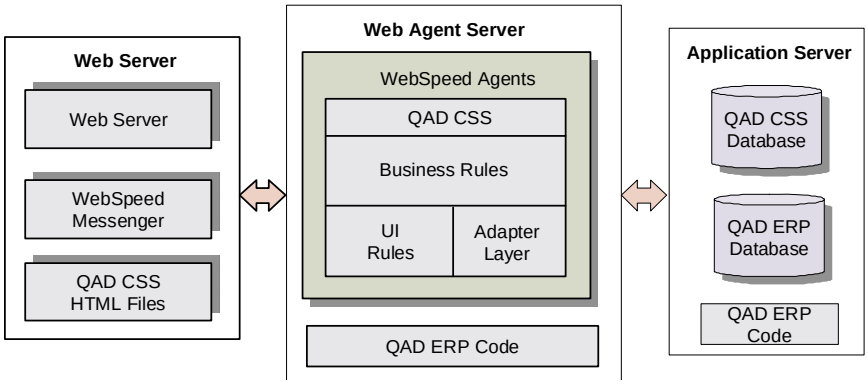
A more common deployment involves separating the Web server from the rest of the product. It is still very efficient for low- to medium-load situations, and is also relatively easy to implement and maintain. Figure 1.3 illustrates an example of a distributed deployment.

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. This 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

Your system should meet 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

You should 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, QAD 2008 Enterprise, QAD 2008.1 Standard, QAD 2008.1 Enterprise, QAD 2009 Standard, QAD 2009 Enterprise, QAD 2009.1 Enterprise, QAD 2010 Standard, QAD 2010 Enterprise, QAD 2010.1 Enterprise, QAD 2011 Standard, QAD 2011 Enterprise, QAD 2011.1 Enterprise, QAD 2012 Enterprise
- OpenEdge OE10.1A, OE10.1B, OE10.1C, or OE 10.2A
- Progress WebSpeed 3.1D and later

  Note If you are completing a multi-tier deployment, WebSpeed must be installed 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

Requirements for Credit Card Processing

If you plan to implement credit card processing, which is required for a B2C order-entry scenario, your Web server has additional requirements. Credit card processing involves a real-time authorization and capture (settlement) of the funds that were authorized. CSS supports the PayPal credit card API. (The Verisign credit card API has been replaced, since VeriSign was acquired by PayPal and continues to maintain the PayPal API only.)

To implement credit card processing, you must complete the following general steps:

1. Obtain an integration kit from PayPal or any other established credit card processing service. The processing service should provide the documentation required to set up the services you need to process transactions. (Other services may require changes to the CSS API.)
2. To accept credit cards over the Internet, you need a special account called an Internet Merchant Account, and you must provide details about this account to your credit card processing service.
3. Execute the steps for setting up credit card processing described in this installation guide.
4. Install and implement Secure Sockets Layer (SSL) on your Web server.
5. Execute the steps for setting up credit card processing described in *Implementation Guide: QAD Customer Self Service*.

Requirements for Integrating with QAD Warehousing

If you want to integrate QAD CSS with QAD Warehousing, which is an optional module within QAD ERP, you must apply two patches—R1R0 and R1R1—to QAD EE.

Preparing to Install

Before you begin the installation process, make sure you know and write down the information about your prerequisite software components as well as your planned CSS installation as listed in Table 1.1. The table also contains naming conventions to reference directory locations that may differ for each installation.

Table 1.1
Preparatory Installation Information

Information	Reference	Description
Progress install directory	*ProgressInstallDir*	The root directory where Progress and WebSpeed are installed on your system
QAD ERP install directory	*QADERPInstallDir*	The directory where QAD ERP is installed
QXtend Outbound install directory	*QXtendOutBoundDir*	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	*WebServerDir*	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
QXtend Adapter directory	*QXtendAdapterDir*	The directory where QXtend Adapter files are installed; for example, `/qad/qad2009/qxtend` 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	*CSSInstallDir*	The directory where you want to install QAD CSS; for example, `/qad/qad2009/qadcss`
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 1.2 lists the components included in the compressed files and summarizes their purpose.

Table 1.2
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 *xx* 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.
`qadcss`	Multiple subdirectories with core product files.
`qadcss/appstart.p`	Additional optional startup options.
`qadcss/appstart.pf`	Additional optional startup options.
`qadcss/Compme3.1.html`	Default compiler program.
`qadcss/qadcss.ini`	QAD CSS configuration file.
`qadcss/qadcss.pf`	Default start parameters for WebSpeed agents.
`qadcss/startcss.p`	Variable setup file.
`qadcss/ systemunavailable.html`	Error file template for WebSpeed errors.
`qadcss/css`	Subdirectories with QAD CSS program files.
`qadcss/css/ad`	Administrative programs.
`qadcss/css/adr`	Administrative report programs.
`qadcss/css/br`	Browser programs.
`qadcss/css/ca`	Feedback and FAQ programs.
`qadcss/css/cc`	Credit card programs.
`qadcss/css/cl`	License clearance programs.
`qadcss/css/ext`	Extend layer programs.
`qadcss/css/he`	Help maintenance programs.
`qadcss/css/lg`	Login programs and programs used to implement the B2C order module.
`qadcss/css/lib`	Common libraries.
`qadcss/css/messages`	Error message program for serious errors.

Table 1.2 — *QAD CSS Components* — (Page 1 of 2)

Directory Component	Content Description
qadcss/css/mfg	Programs that directly interface with QAD ERP database.
qadcss/css/op	Order processing programs.
qadcss/css/opr	Order processing report programs.
qadcss/css/rep	Replication templates for populating items. **Note**: Catalog Load and Customer Load are the preferred methods for loading this data.
qadcss/css/sys	System programs and base report programs.
qadcss/css/tools	Utility programs.
qadcss/css/tt	Temp table definitions used with all programs.
qadcss/images	QAD CSS default images.
qadcss/images/eng	Images for the English language.
qadcss/images/*xx*	Images for non-English languages; *xx* refers to the QAD ERP language code.
qadcss/images/items	Images for items in the catalog. Initially empty.
qadcss/images/oe_icons	Images for action icons. Initially empty.
qadcss/logs	WebSpeed server logs. Empty when created.
qadcss/scripts	Java scripts used by QAD CSS.
qadcss/scripts/eng_msg	English-based JavaScript messages.
qadcss/scripts/*xx*_msg	JavaScripts for non-English languages where *xx* refers to the QAD ERP language code.
qadcss/scripts/menu	Menu JavaScripts.
qadcss/styles	Default cascading stylesheets used by QAD CSS.
qadcss/temp	QAD CSS temporary files. Empty when created.
qadcss/temp/email	Used to process e-mail events.
qadcss/tools	Template and sample files.
qadcss/upload	Workspace for loading customers and catalogs during implementation.
qadcss/web	WebSpeed agent extension files.
qadcss/web/objects	Progress WebSpeed files.
qadcss/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 1.2 — *QAD CSS Components* — (Page 2 of 2)

Chapter 2

Installing QAD CSS

Use the instructions in this chapter to install and configure QAD CSS on UNIX, Linux, or Windows platforms. If you are upgrading an existing QAD CSS installation to the latest release, use the instructions in Chapter 3 rather than these instructions.

Overview

Use the instructions in this chapter to install and configure QAD CSS on a UNIX, Linux, or Windows platform.

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 the installation user has write and execute permissions for the target installation directories. QAD recommends creating a user `mfg` in the group `qad`. This should already have been done as part of the QAD ERP installation.

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 2.1.

Table 2.1
UNIX CD Drive Mount Commands

Hardware	Mount Command
Sun	`volcheck cdrom`
HP	`/etc/mount -F cdfs /dev/dsk/`*`YourCDDevice`*` /cdrom`
Digital	`mount -r -o noversion -t cdfs /dev/`*`YourCDDevice`*` /cdrom`
AIX	`smitty mountfs` Then select file system, directory, and file system type (`cdrfs`).
Linux	`mount /dev/hdb /mnt/cdrom` Where `/hdb` could be `hdc` or `hdd` among other possibilities.
All others	Refer to your operating system documentation or vendor for requirements to mount a CD-ROM. You may be able to type `man mount` 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 2.2 to enter the appropriate values for script execution.

Table 2.2
Install Script Steps

Step	User Values
License agreement	Yes
Install log file	Accept default or enter a new location and name
Create `instcss.ini`?	Yes
QAD CSS install directory	The directory where you want to install QAD CSS; for example, `/qad/qad2009/qadcss`.
For what language?	Simplified Chinese (ch)
	Castilian Spanish (cs)
	Dutch (du)
	French (fr)
	German (ge)
	Italian (it)
	Japanese (jp)
	Latin Spanish (ls)
	Portuguese (po)
	Traditional Chinese (tw)
	English (eng)
Are you running 2012 EE edition of QAD Enterprise Application?	Yes
Install summary	Yes
File extraction	None
Progress directory	Your Progress install directory
Windows: icon folder name	Default is CSS
	5.2
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 `qadcss` 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 this:

```
$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 used by 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 used by your QAD ERP database as the value of the -cpinternal parameter, but 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 Czech	1250
Korean	EUC-KR
English Dutch French German Italian Latin Spanish Castilian Spanish Portuguese	ISO8859-1

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

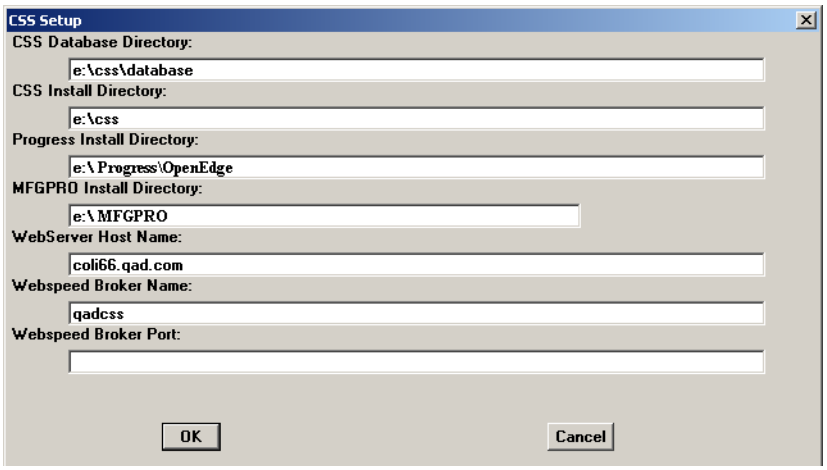


Table 2.3
CSS Setup Fields

Field	Description
CSS Database Directory	This is *CSSInstallDir*/database by default.
CSS Install Directory	The *CSSInstallDir* defined during install.
Progress Install Directory	The Progress install directory as identified during the install.
MFGPRO Install Directory	The *QADERPInstallDir* for the QAD ERP instance CSS will connect with.
WebServer Host Name	The server name or IP address of the Web server. This should also include a domain name. If the domain name is not included, the data entered in the screen may not be saved properly. The .qad.com in the sample screen is the domain name.
WebSpeed Broker Name	The name of the broker you plan to use; this guide uses qadcss as an example.
WebSpeed Broker Port	An available port specified as a part of the “Preparatory Installation Information” on page 7.

4 Choose OK to save the changes.

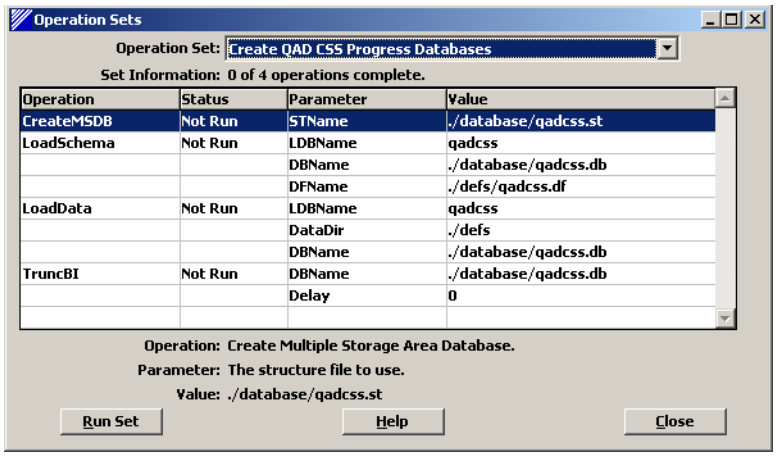
Create the QAD CSS Database

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 the set up.

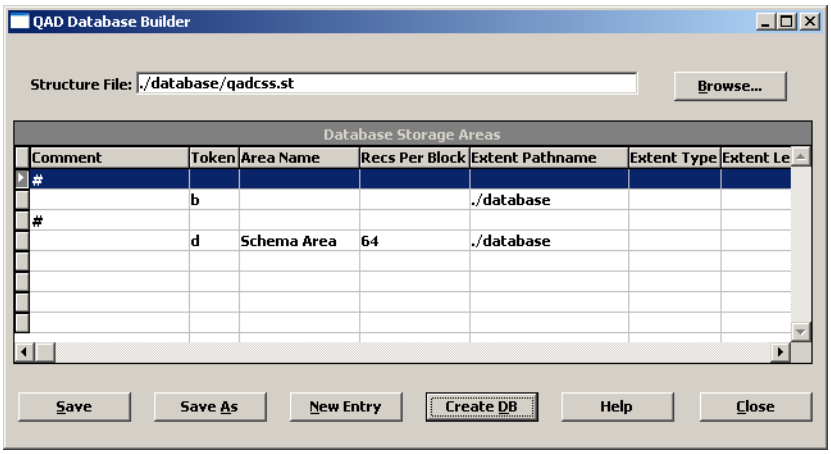
Fig. 2.2
CSS Guided Setup Screen



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. You may want to edit the default values as you proceed.

4 The QAD Database Builder opens displaying the contents of `qadcss.st`. This is the `qadcss` database structure file. You can edit the structure file in this screen.

Fig. 2.3
qadcss Structure File



5 Choose Create DB to create the new database.

6 You are then asked what database to copy to create the new database. Choose Other Database and enter the path to the empty database with the same code page used by your QAD ERP databases.

Progress OpenEdge supplies a collection of empty databases in a variety of 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.

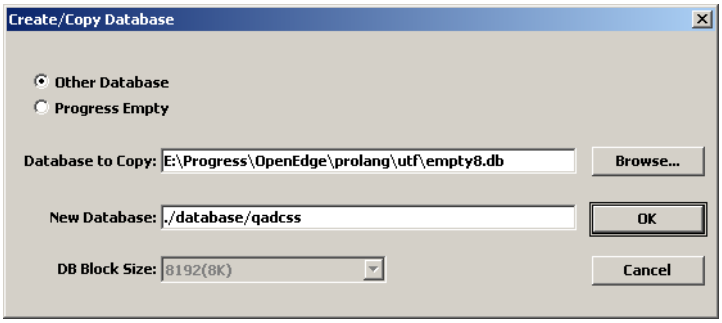
To create a database that uses the utf-8 code page, copy the following empty database:

ProgressInstallDir`/prolang/utf/empty8.db`

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

Note The DB block size must be set 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

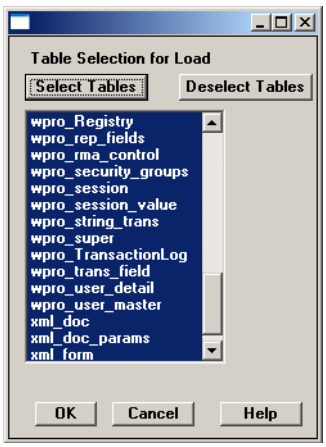
10 The correct file name and location appear by default. Choose OK. The schema loads and a log window displays the progress. Choose Close to shut the log window and continue.

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.

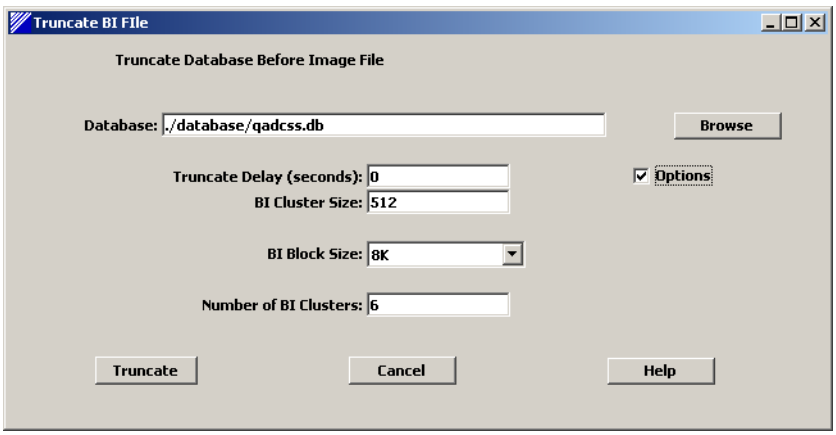
Fig. 2.7
qadcss Table Selection for Load



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_se/mfg
```

```
./css_se
./css/mfg
./css
../EDI_API
../EDI_API/us
../mfgpro/eB21/QADERPSPVer
QXtendAdapterDir
QXtendAdapterDir/triggers
QADERPInstallDir
QADERPInstallDir/us
QADERPInstallDir/triggers
QADERPInstallDir/xrc
QADERPInstallDir/bbi
```

Note Add *QXtendAdapterDir* and *QXtendAdapterDir*/triggers to the PROPATH only when the QXtend Outbound events database is installed.

- For QAD EE:

```
./
./css_ee/mfg
./css_ee
./css/mfg
./css
QXtendAdapterDir
QXtendAdapterDir/triggers
QADERPInstallDir
QADERPInstallDir/bbi
QADERPInstallDir/fin
QADERPInstallDir/triggers
QADERPInstallDir/xrc
QXtendOutBoundDir/xrc
QADERPInstallDir/Configs
```

Note Add *QXtendAdapterDir* and *QXtendAdapterDir*/triggers to the PROPATH only when the QXtend Outbound events database is installed.

QXtendAdapterDir. QXtend Adapter directory specified as a part of the “Preparatory Installation Information” on page 7.

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

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

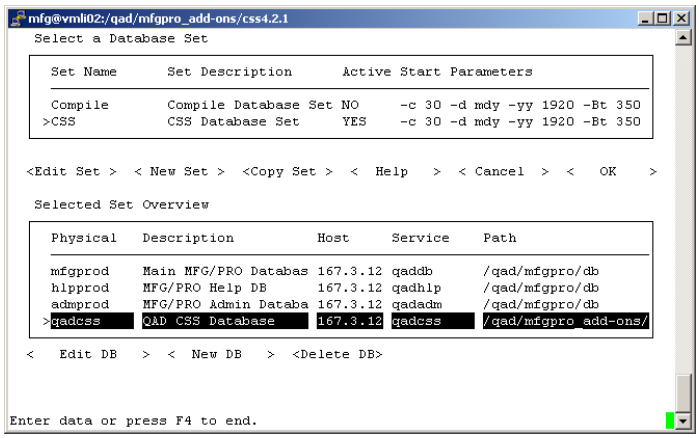
QADERPSPVer. The service pack version number of your QAD ERP; for example, SP8.

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

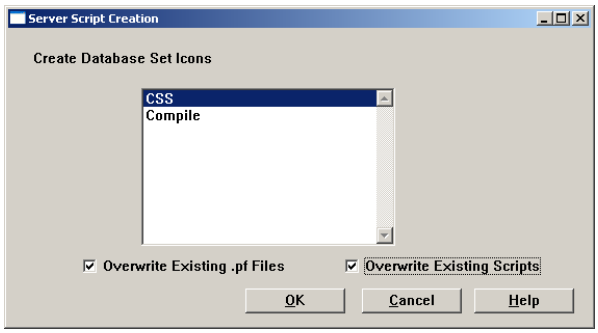


5 When finished, choose OK to save the settings.

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 used by 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/qadcss -L 1000 -B 5000 -cpinternal CP936 -cpstream CP936
```

- stopdb

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

Note When setting the code page parameters:

- For QAD SE and earlier versions, specify the same code page used by 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 used by your QAD ERP database as the value of the -cpinternal parameter, but 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 Dutch French German Italian Latin Spanish Castilian Spanish Portuguese	ISO8859-1

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

Use an available port to be used by the CSS database server specified as a part of the “Preparatory Installation Information” on page 7.

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

```
#EXAMPLE ENTRY:
# <service_name>      <portnumber/protocol>  #<description>
qadcss                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

The QAD CSS database needs to be run in multi-user mode to allow 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 you execute this sequence successfully. The QAD CSS database must be started 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 help.

1. Verify that the AdminServer process is running.
   - If the software is located 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 located on a UNIX server, run the following command on that server:

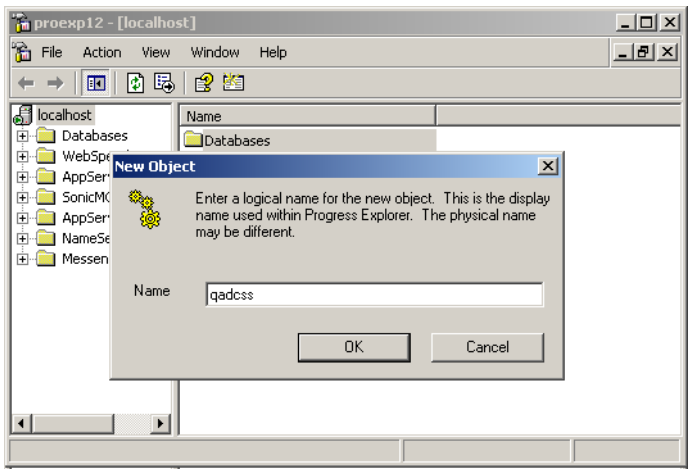
     ProgressInstallDir\bin\proadsv -query

     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 new 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 this 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 `qadcss` and click OK.

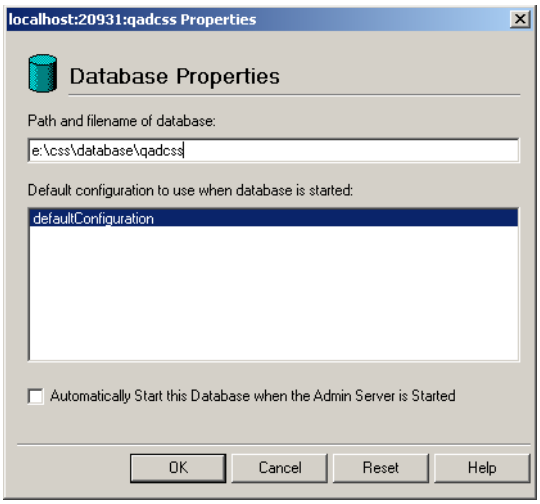
Fig. 2.11
Adding the qadcss Database



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

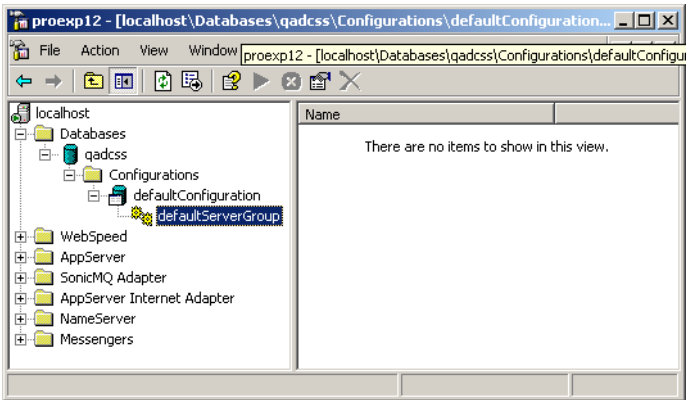
Note The `.db` extension for the database file is not required.

Fig. 2.12
Database Properties



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

Fig. 2.13
Expanding Configuration Node



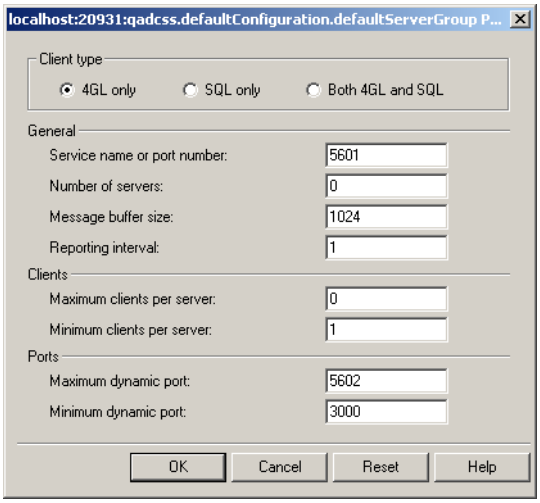
8 Right-click defaultServerGroup and choose Properties. In the dialog box:

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.

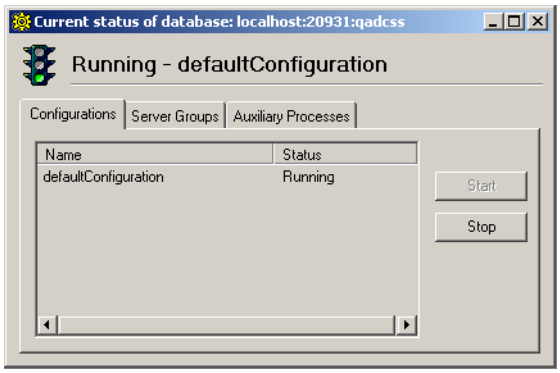
Fig. 2.14
Database Configuration Properties



9 Right-click the database name in the left window and choose Start.

10 Right-click the database again and choose Status. The database should be running.

Fig. 2.15
Database Running



Setting Up WebSpeed

This section describes how to create a new 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 31.

This procedure assumes you are installing all QAD CSS components on one server. In a multi-tier installation, you may need to 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 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 on 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.

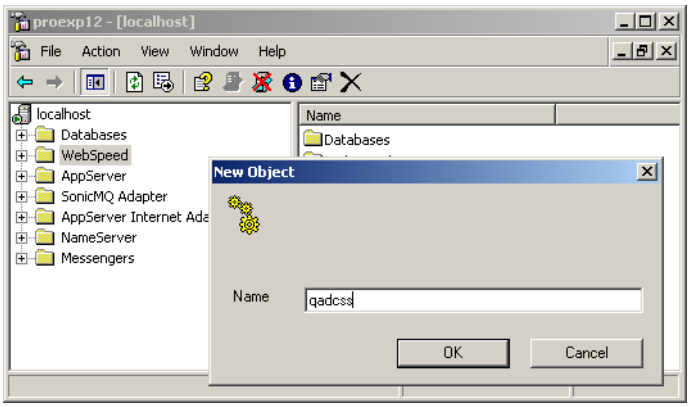
   The Progress Explorer connects to the AdminServer process running on the machine where you are configuring the WebSpeed instance. A connection is indicated by a green arrow in the machine icon.

Set Up the WebSpeed Broker

1 Once you have connected to the remote or localhost server machine, click on 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.

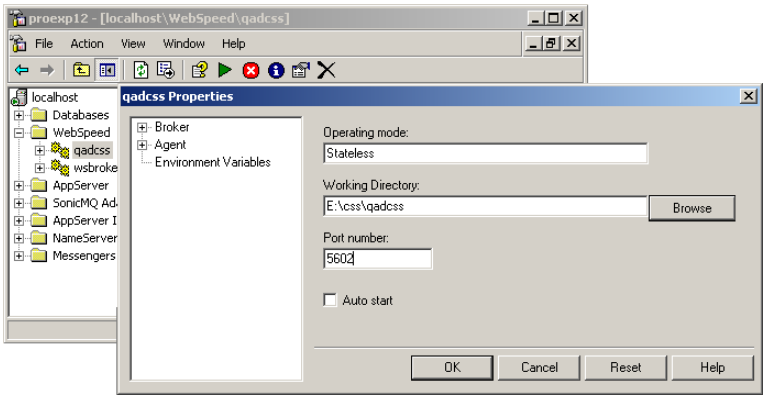
3 Expand the WebSpeed directory; right-click the broker that was just created and choose Properties. A Properties dialog box displays.

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 Installation Information” on page 7.

c Leave the Operating Mode set to Stateless and Auto start set to No.

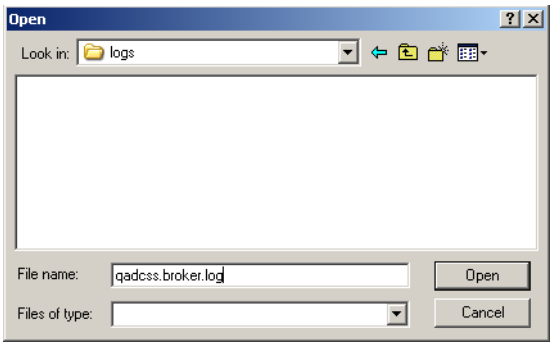
Fig. 2.17
WebSpeed Broker Properties



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`*`\qadcss`. 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

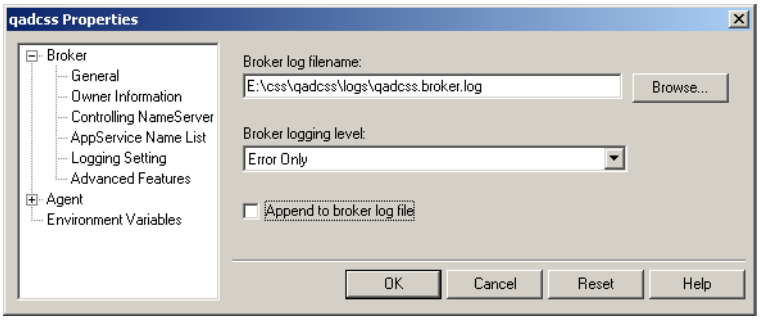


b Click Open.

c Choose Error Only in the Broker logging level drop-down list.

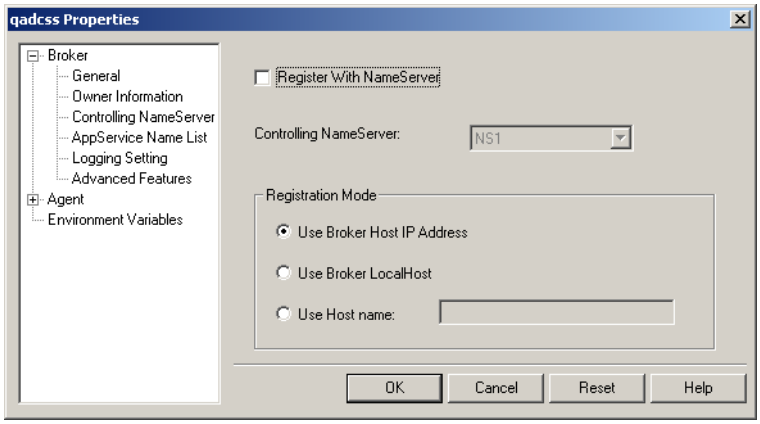
d Uncheck Append to broker log file.

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, uncheck Register with NameServer.

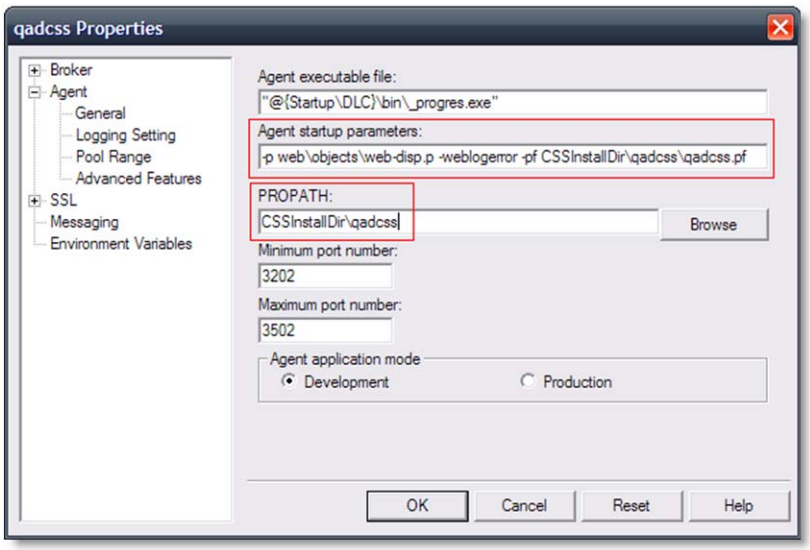
Fig. 2.20
Unregistering the NameServer



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



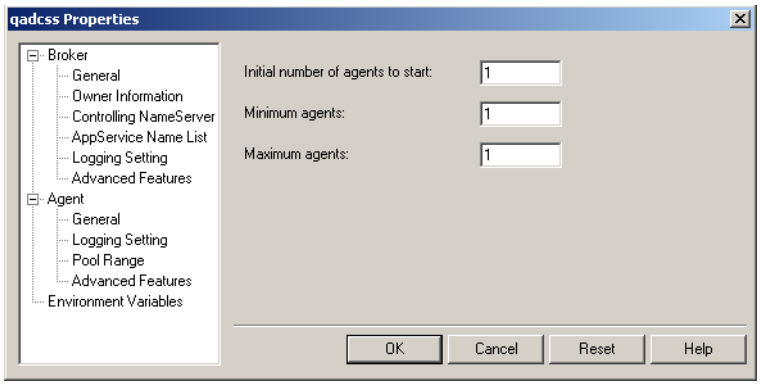
Agent startup parameters. Enter the following:

`-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



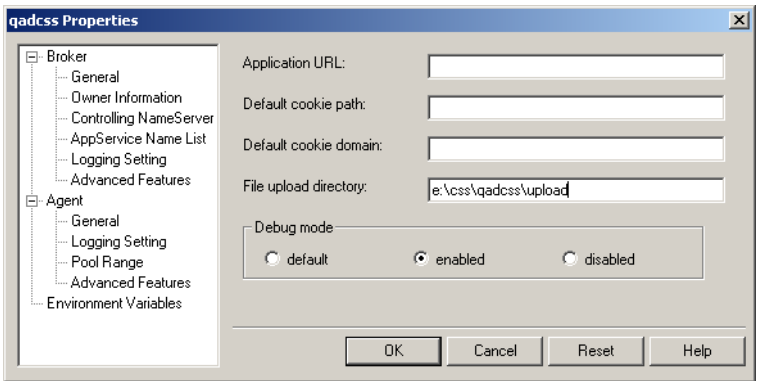
Note After you register your QAD CSS clearance code, return to this section and update the pool settings based on the licensed number of agents.

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 should be:

CSSInstallDir\qadcss\upload

Fig. 2.23
Entering the File Upload Directory



11 Choose OK to complete the configuration.

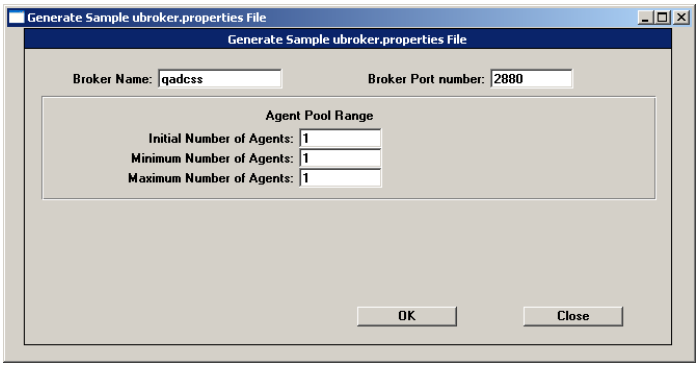
12 Choose File|Exit to exit the Explorer tool.

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 Leave the agents all set to 1. Later, after implementation, you will need to reset these 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`\qadcss-ubroker.properties`

The file should look something like the following:

```
[UBroker.WS.qadcss]
    appserviceNameList=qadcss
    brokerLogFile=CSSInstallDir\qadcss\logs\qadcss.broker.log
    srvrLogFile=CSSInstallDir\qadcss\logs\qadcss.server.log
    description=qadcss
    environment=qadcss
    controllingNameServer=NS1
    portNumber=3601
    PROPATH=CSSInstallDir\qadcss
    uuid=6fcd1c8e29b18fd3:55af5:ff79d89dd3:-8000
    workDir=CSSInstallDir\qadcss
    fileUploadDirectory=CSSInstallDir\qadcss\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\qadcss\qadcss.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`
- `qadcss.ini`

3 Choose Close to close the log window.

4 If you have installed the QXtend database, edit *CSSInstallDir*/qadcss/qadcss.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*/qadcss/qadcss.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*/qadcss/qadcss.pf file and check for the -cpinternal, -cpstream, and -cpcoll parameters. Make sure you use the same code page and collation used by the QAD ERP database; for example:

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

7 If you are using QAD EE versions earlier than 2011 EE, copy the *CSSInstallDir*/qadcss/css_ee/pre2011EE/mfsubdirs.i file to the *CSSInstallDir*/qadcss/css_ee directory and overwrite the existing file.

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

1 Restart the QAD CSS service (in this document, qadcss) 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*/qadcss/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*/qadcss/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 may have 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*/qadcss are moved to a directory on the Web server. You need to create an alias for this directory.

Important In a production environment, do not alias the full *CSSInstallDir* directory because this 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 /qadcss/ "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 Microsoft's Internet Information Server (IIS):

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, qadcss.
5. Enter the path to the directory that contains the `/images`, `/scripts`, and `/styles` subdirectories.

Install and Configure WebSpeed Messenger

If you do not have a full WebSpeed installation on your Web server in a multi-tier installation, you must 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 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 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 `msngrs.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.

   ```
   msngrExecFile=@{Startup\DLC}\bin\wspd_cgi.ksh
       AllowMsngrCmds=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

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

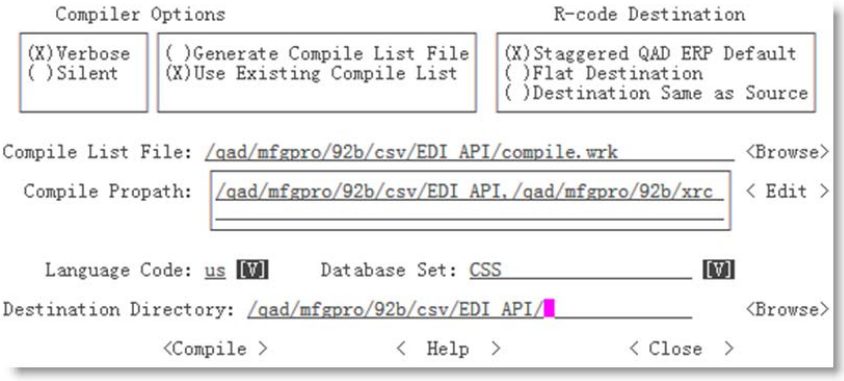
Windows server

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

Applying Patches

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

Fig. 2.25
Compile EDI_API



For QAD 2009 EE and all earlier versions, apply the R1PM patch. For information on this patch, see solution ID qad72463 in the QAD KnowledgeBase.

For QAD 2010 EE and all earlier versions, apply the R1ZR patch.

For QAD 2011 EE and all earlier versions, apply the MFG-6398 patch.

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 `qadcss.ini`, the QAD CSS WebSpeed broker must be restarted.

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 may take 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 those in the furnished example scripts, located in *`CSSInstallDir`*`/utils_ux`. In the example, they have been copied to *`CSSInstallDir`*.

The WebSpeed broker may need 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 in order 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.

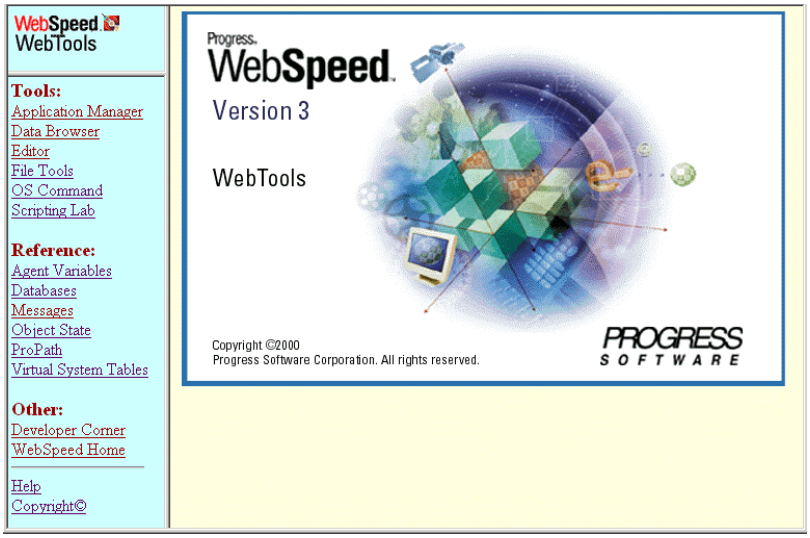
1 Using a Web browser, go to the WebSpeed Workshop by entering the following URL:

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`.

A screen similar to the following should display.



2 Click the Databases link from the menu.

A drop-down list displays the connected databases.

Important The QAD CSS and QAD ERP database must be connected in order to continue with the installation.

Verify PROPATH

Because you must update the PROPATH as part of the configuration process, you should 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 `qadcss.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

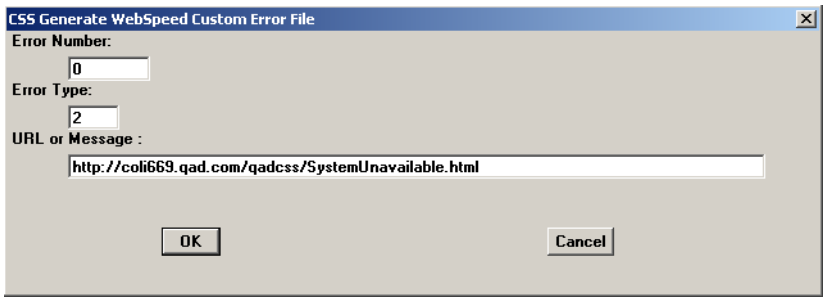
You now create a text file that redirects WebSpeed errors to a URL.

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 will appear 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, you must 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



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

   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.

4 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:

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

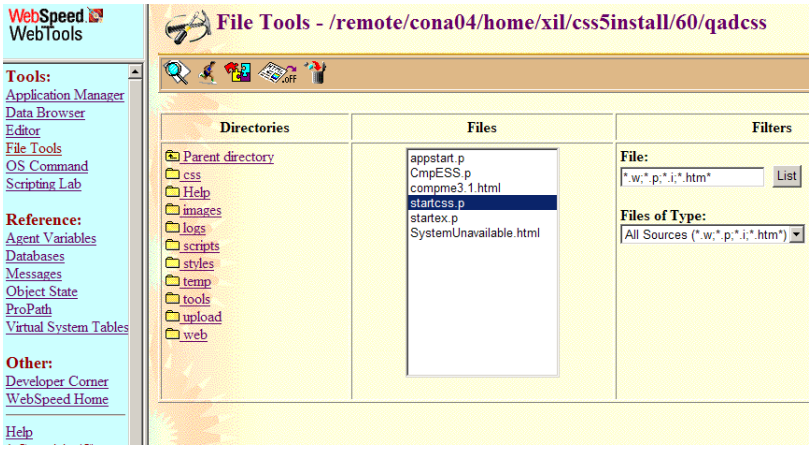
http://*webServer*/*webServerScriptsDirectory*/wspd_cgi.sh/WService=*brokerName*/workshop

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

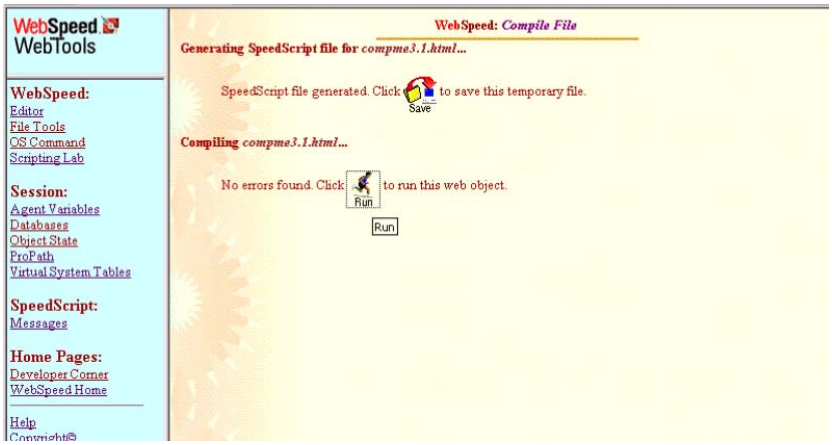
Note In Windows environments, `cgiip.exe` is typically specified rather than `wspd_cgi.sh`.

2 Select File Tools from the menu on the left.

3 Compile `compme3.1.html` in the WebSpeed Workshop.



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



5 In Filter, enter `.html`. Highlight all files to be recompiled 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 these are loaded persistently, compiling them does not improve performance significantly.

The compilation process may take 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.

Note For QAD SE and earlier versions, do not compile the file `css_ee/mfg/lib/upsell.html`.

6 Compile the two `.w` files in *`CSSInstallDir`*`/qadcss/css/cl/`. In Filter, enter `.w`. Highlight all files to be recompiled 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.

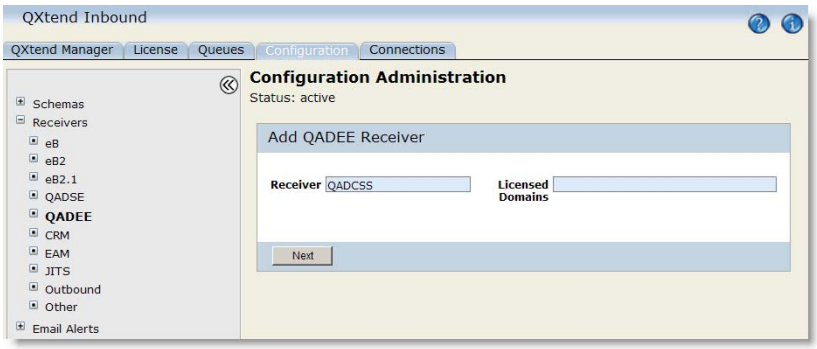
a In Configuration Manager:

- 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 1.1	ERP3_1	UI API Adapter	adcsmt.p	adcsmt-ERP3_1.xml

- QAD SE and earlier versions:

QdocName	XML Syntax	Version	Route	Procedure	Event
deleteCustomerShipTo	QDoc 1.0	eB_1	UI API	adcsmt.p	adstmtd-eB_1.xml
deleteCustomerShipTo	QDoc 1.1	eB_2	UI API	adcsmt.p	adstmtd-eB_2.xml
maintainCustomer	QDoc 1.0	eB2_1	UI API	adcsmt.p	adstmtd-eB2_1.xml
maintainCustomer	QDoc 1.1	eB2_2	UI API	adcsmt.p	adstmtd-eB2_2.xml
maintainCustomerShipTo	QDoc 1.0	eB2_1	UI API	adcsmt.p	adstmtd-eB2_1.xml
maintainCustomerShipTo	QDoc 1.1	eB2_2	UI API	adcsmt.p	adstmtd-eB2_2.xml

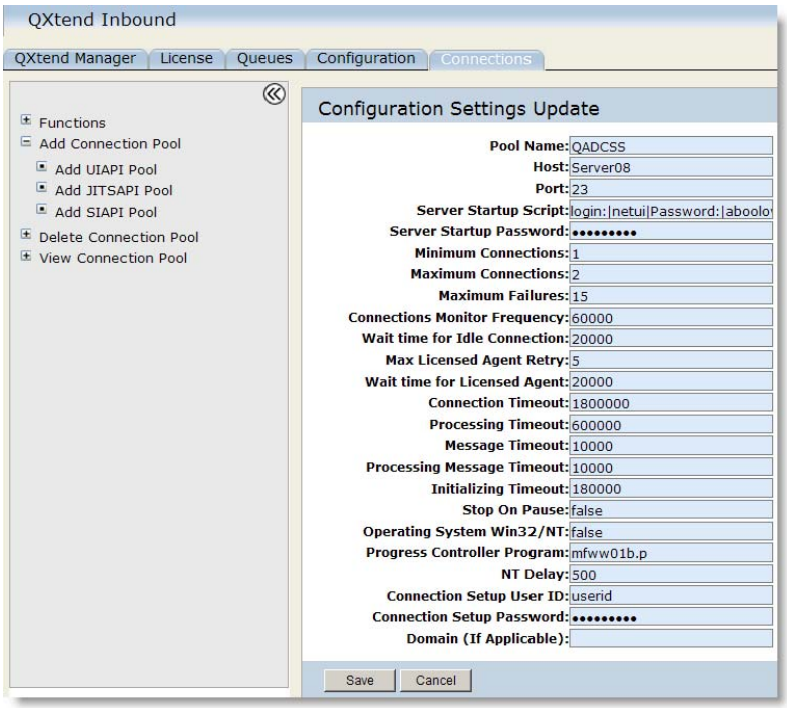
f A summary page is displayed. The receiver has been successfully added.

4 Go to Connection Pool Manager and create a UIAPI connection pool for the receiver just 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 need to add a database connection to the CRM database in the UIAPI connection setting.



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|$PASSWD|osPrompt| startScript. For example:

`login:|`*`QXtend`*`|Password:|`*`password`*`|$|exec ./qma.QXprod.`

For Oracle implementations, the qma script must be modified.

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 $PASSWD reference.

Minimum Connections. Enter the minimum number of open connections that the Connection Pool Manager should maintain. 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 should allow. The Connection Pool Manager will not open any more connections than this.

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

Maximum Failures. Enter how many times the Connection Pool Manager should attempt 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). This monitors all connections in all states and closes those 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. The maximum number of connections may have been reached, or new connections may be in the initializing state. The default value is 20000 (20 seconds).

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

Wait Time for Licensed Agent. Specify the number of milliseconds that the system will wait 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 this.

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, this should always be set to false. This prevents a transaction from failing when a “Press Spacebar” message is displayed in the target QAD ERP session.

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

Progress Controller Program. Enter `mfww01b.p` for UI API pools.

NT Delay. This can safely be ignored for QXI connection pools.

Connection Setup User ID. This 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

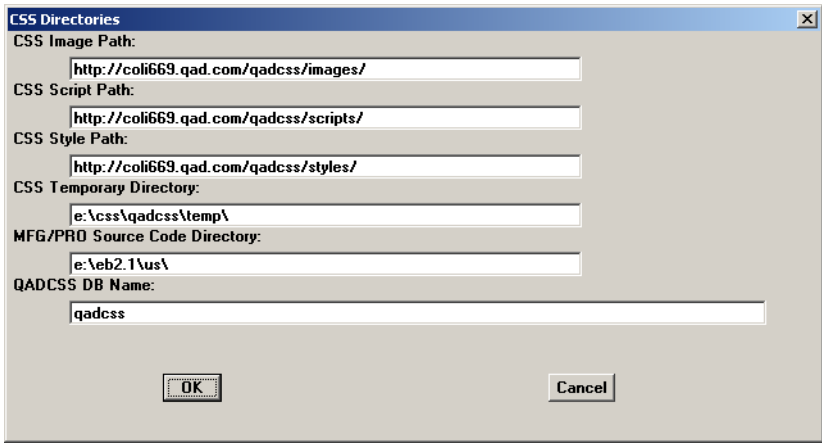
Before you can access the QAD CSS database, you must verify your system variables. Later, these variables can be changed from administrative functions on the QAD CSS menu.

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 30, do so now; otherwise, the connect will fail.

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/qadcss/images/
   - A relative URL, such as /qadcss/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/qadcss/scripts/

- A relative URL, such as /qadcss/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/qadcss/styles/
- A relative URL, such as /qadcss/styles/

Temporary Directory. The directory identified in your `qadcss.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.

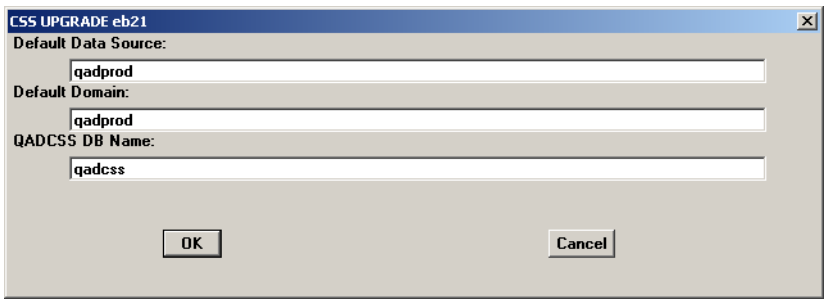
Choose OK to save any changes and write the directories to WebSpeed.

Set the Default Data Source

When implementing QAD CSS with QAD ERP, you must define a default data source in order 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



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, `cgiip.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 will be 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 to determine which should be changed to optimally configure the QAD CSS application. Initially, you can accept all defaults. However, note the following fields:

- Temp Directory should reflect the appropriate path to the `/temp` directory that will be used by the application.
- Extend to ERP should always be selected 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 should be *`myQdocWebService`* configured in QXtend, such as `http://167.3.129.36:8080/qxi1.6/services/QdocWebService`.

  For information about configuring QXtend Inbound, see *Technical Reference: QAD QXtend*.
- Help Path should be set to the virtual directory on the web server where the QAD CSS Help files reside; for example: `/qadcss/Help/`.

Note After you change the control settings, the system must be restarted 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 Adminstration|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 this is not a valid user in the QAD ERP system you will be accessing, change it now.

d Repeat these steps for QADUPasswd to secure this user with a password.

Important You must also 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 again for QADUDomain, entering the default data source set up 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*/qadcss/scripts.

Note In a multi-tier installation, when the Web server is a Windows server and the application server is on UNIX or Linux, you need to 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. This will compile a JavaScript file containing messages required by the system.

Note During use of the application, you need to do this whenever you add a new 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 redisplays 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; you will return to the menu in the next procedure.

Update QAD ERP for Credit Card Processing

If you plan to implement credit card processing, you must 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.

Note Make sure JRE 1.4 or later is installed before applying the patch. JRE can be downloaded from www.java.com.

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

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`. It should contain 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. This file must be located in the QAD ERP PROPATH. Similar settings are defined in QAD CSS in Credit Card Vendor Code Maintenance.

Important You must ensure that the settings in the two places are the same or errors may occur while processing credit card orders.

Here is a sample `verisign.ini` file.

```
[Verisign]
VendorHost=pilot-payflowpro.paypal.com
VendorPort=443
TimeOut=30
Partner=VeriSign
Vendor=QADCSS
User=QADCSS
Password=123
VendorUrl=/transaction
CssJarHome=/qad/mfgpro/93/csu/cust/payflowpro/linux
JavaRunnableWithPath=/users/johndoe/jdk1.6.0_10/bin/java
```

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 Verisign, typically 443. This is used by the API when transactions are sent to be authorized.

Partner. Specify VeriSign.

Vendor. Verisign 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. This is typically the same as Vendor.

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

CSSJarHome. Specify the path to the PayPal processing program jar file.

JavaRunnableWithPath. Enter the full path of the Java executable file from the JRE installation.

Update the CSS system registry as follows:

Module	Override	Key	Key Value
sys	No	JavaPath	Change the value of the registry key JavaPath to the path of the JRE java executable. Do not include the last path separator (/ for Linux and \ for Windows). If JRE is installed under */usr/jdk1.6.0_10*, then the value of the registry should be `/usr/jdk1.6.0_10/bin`.
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 Host Name to pilot-payflow.paypal.com for the test environment or payflow.paypal.com for the production environment.
- Change Path to a Processing Program directory. The default is: *`CSSInstallDir`*`/qadcss/tools`.

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

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

Prerequisite: Make sure QAD Configurator has already been installed.

1. In the *CSSInstallDir*/qadcss/css/cfg directory, open the config.i file and set *configurator-integrated* to Yes.
2. In the *CSSInstallDir*/qadcss directory, open the qadcss.ini file and add *CPDInstallDir*, *CPDInstallDir/LanguageCode*, and *CPDInstallDir*/cop_xrc to the PROPATH; also add the database of QAD Configurator in the qadcss.ini file.
3. Compile the QAD CSS source code.
4. In System Control, set *Integrated with Configurator* to Yes.
5. Use System Registry Maintenance to specify the value of the registry *ConfiguratorWSURL* as http://*cfgWebSpeedServerHostName*/cgi-bin/wspd_cgi.ksh/WService=*cfgWebSpeedBrokerName*

Final Steps

After you have successfully completed the installation and verified the setup, you complete these 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 licensed number of agents you have registered and click OK.
3. Complete QAD CSS implementation using *Implementation Guide: QAD Customer Self Service*.

Chapter 3

Upgrading QAD CSS

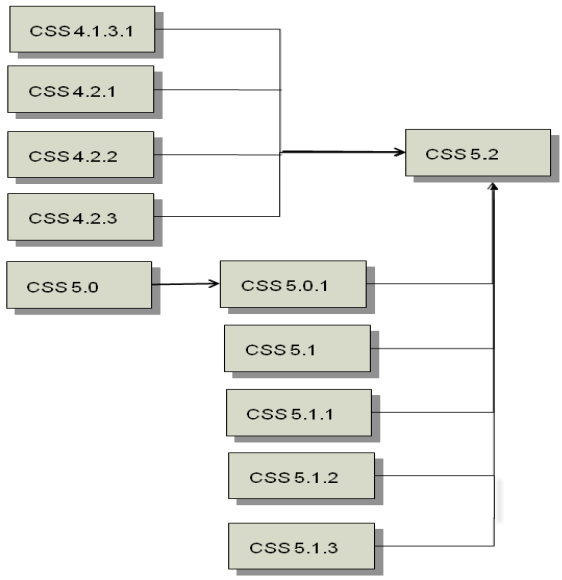
Use the instructions in this chapter to upgrade from previous QAD CSS releases to 5.2 on UNIX, Linux, or Windows platforms.

Overview

Use the instructions in this chapter to upgrade from previous QAD CSS releases to 5.2 on UNIX or Windows platforms.

The conversion paths for different CSS versions to be upgraded to 5.2 are illustrated in Figure 3.1:

Fig. 3.1
QAD CSS Conversion Paths



Except for CSS 5.0, which needs to be upgraded to CSS 5.0.1 first, all earlier versions of CSS can be directly upgraded to CSS 5.2 using the following general steps:

- Upgrading Progress to OpenEdge 10.1A
- 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 *Installation Guide: QAD CSS 5.0.1*.

Upgrading CSS to Version 5.2

Upgrading Progress to OpenEdge 10.1A

If your existing QAD CSS system uses an earlier Progress version, upgrade it to OpenEdge10.1A 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 12.

Setting Up QAD CSS Environment Values

Before beginning the conversion, you must define some environment variables required by MFG/UTIL.

Follow the same steps in “Set Up QAD CSS Environment Values” on page 13.

Generating Database Scripts

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

Follow the same steps in “Generate Database Scripts” on page 18.

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 5.2 install directory.
2. If you upgrade your Progress version to OpenEdge 10.1A, you must 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, you must edit file `mfgutil` to change the following parameters:

   `-cpstream` *`NewCodePage`* `-cpinternal` *`NewCodePage`*
4. If you are upgrading from 4.1.3.1 through 4.2.3, load an incremental 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 QAD CSS 5.2.

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 QAD CSS 5.2 release.

9 For QAD CSS versions prior to 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.

10 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 procedure to convert your existing QAD CSS database.

Fig. 3.2
Converting Database

1 Copy the database to be upgraded from its existing location to the `/database` directory under your QAD CSS 5.2 installation.

   a Change your current directory to *`CSSInstallDir`*.

   b Use `procopy` to copy your previous version database to:

   `CSSInstallDir``/database/qadcss`

2 If you upgraded your Progress version to OpenEdge 10.1A, you must 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 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, if you are upgrading from QAD CSS 4.2.1, choose:

      `CSSInstallDir``/defs/upgrade/`*`4.2.1`*`/delta421_52.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, you need to load two files—`/upgrade/4.2.2/delta422_423.df` and `/upgrade/4.2.3/delta423_52.df`.

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

4 Start MFG/UTIL and choose Convert CSS from the Upgrade CSS menu.

   a In the CSS Upgrade Original Version window, enter the original QAD CSS version number you are upgrading from; for example, 5.1.1.

   b In the Connect Database screen, connect to the existing CSS database to be converted (the one copied in step 1). 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 this is complete, choose Close to continue.

5 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 this is complete, choose Close to continue.

6 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`.

7 Choose Update Product and Version from the Upgrade CSS menu.

   a Enter the correct version name and database name. The version name for the current version is `ex 5.2`. The database is the QAD CSS database name located in:

      `CSSInstallDir``/database`

   b The Log window appears and shows progress. Choose Close to continue.

   Note When updating licensing records, it is possible that errors regarding duplicate records may display. You can safely ignore these records.

8 If you are upgrading from QAD CSS 4.2.1, skip to step 9.

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 this is complete, choose Close.

9 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 upgraded your Progress or WebSpeed version as part of your QAD CSS migration, you will need to tailor the configuration of these products for QAD CSS. Follow the instructions in “Installing QAD CSS” under:

- “Setting Up WebSpeed” on page 25
- “Configuring QAD CSS” on page 32

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 46.

Final Steps

Follow the same steps in “Set Up Integration of QAD CSS and QAD Configurator (Optional)” on page 52.

Index