Administration Guide

QAD Planning and Scheduling Workbenches

Introduction
Workbench Processes
System Load and Performance
Configure

70-3120A
QAD Planning and Scheduling Workbenches
QAD EE 2011
March 2011

PlanSchedWrkBench_AG_v2011EE.pdf/crl/crl

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

Contents

Chapter 1

Introduction

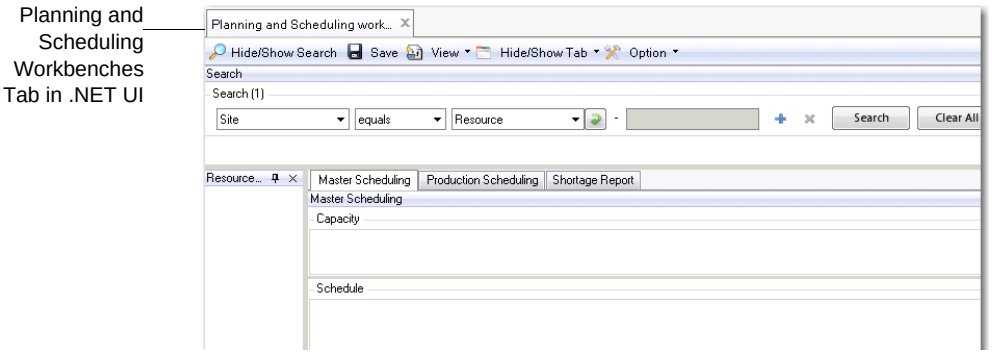
This chapter introduces Planning and Scheduling Workbenches and its administration functions.

Overview

The QAD Planning and Scheduling Workbenches are .NET UI-only scheduling tools that run in the QAD Enterprise Edition. The tool lets you effectively plan and schedule resources for a master schedule or a production schedule from a single workbench for each type of schedule.

You run the workbenches from the .NET UI's list of applications. The .NET UI tab indicates the Planning and Scheduling Workbenches; see Figure 1.1.

Fig. 1.1
QAD Planning and Scheduling Workbenches



Once you access the workbench, you can use the following functions and features:

- Master Scheduling Workbench (MSW)
- Production Scheduling Workbench (PSW)
- Integrated component check capabilities
- Several QAD EE programs and browses that provide supporting information
- The flexibility to modify and configure virtually every aspect of the UI to reflect the needs of each application user in your organization

This document focuses on topics of interest to system administrators who troubleshoot issues or problems, analyze behavior, interact with the components or processes, or customize the workbenches. Topics within this guide include discussions on:

- How the workbenches work and background processing
- System load and performance and troubleshooting information
- Data dimensions
- XML that builds the UI
- UI customization

The following topics briefly discuss each major component; for more information, see U*ser Guide: Planning and Scheduling Workbenches*.

Master Scheduling Workbench (MSW)

The MSW increases master scheduling efficiency by letting you simulate and commit scheduling changes, factoring demand, supply, inventory, production orders, and MRP data from several QAD EE programs in a single workbench.

Note Production orders are all orders associated with production—production lines, work centers, or other production areas—including discrete orders, repetitive orders, scheduled orders, cumulative orders, and so on.

You can use the MSW to interact with production line and work center schedules and make changes where necessary. Within MSW, you can update the production order status, as well as release, create, or close production orders, while considering all supply, demand, and capacity sources from the single workbench. You can also identify items with demand issues and check component availability for each production order to be released.

User-configurable parameters control the number of days that you are in control of the schedule as opposed to MRP control. You can also set the number of future and historical days to which you want visibility to your production data in the MSW.

Event-based color coding lets you easily identify areas of concern. You can review and manipulate schedule and production order data. Once satisfied, saving your schedule creates firm repetitive production schedules or revised and new production orders in QAD EE applications.

Production Scheduling Workbench (PSW)

Once you generate a master production schedule over a daily, weekly, or monthly horizon, you may need to create a production schedule for a shorter, two-to-five day period for the shop floor to drive shop floor execution. The production scheduling goal is to optimize shop floor efficiency by scheduling and sequencing production orders that have like attributes together, reducing machine setups and maximizing labor utilization.

Some companies run a single production order over several days, while others run multiple production orders within a single day. Further, some companies define a production sequence/priority by shift to monitor shift performance or to ensure that products are available for a specific shipment time. The PSW lets you schedule discrete and repetitive items on production lines. You can schedule, sequencing items within a day and shift.

MSW and PSW License

Like Operational Metrics, MSW and PSW require an active Maintenance license before you can access them. You obtain the Maintenance license key through the following Web site:

https://support.qad.com/license_keys/activemaintenance/

In the Web URL, click the Generate License Key button to generate a license key, then follow the prompts to generate the license key. Once you select the Accept key, the system generates the key, displaying it on the Web URL screen, and e-mails the license key to you.

Note The Support Web site requires your Support login information.

Once you obtain the license key, you must register the Maintenance license with the system through License Registration (36.16.10.1).

If you attempt to run either the MSW or PSW and you do not have an active Maintenance license, the system displays an error message. The system displays a warning if the active Maintenance license is close to the license expiration date. Refer to *User Guide: QAD Security and Controls* for information on registering the license key in QAD EE. Refer to the *QAD .NET User Interface 2.8.1 Release Notes* for information on Operational Metrics Licensing.

Conversion for Planning and Scheduling Workbenches

A conversion program for the workbenches automatically installs and runs during the standard QAD EE upgrade process. There is no user interaction required for the conversion program.

The program examines existing production line/item detail (lnd_det) records and sets the Run Crew Size field to 1 (one) if it is currently 0 (zero). When you enable the Run Crew Size field for the workbenches calculations, the system requires that the value of Run Crew Size be set to 1 to avoid issues that can arise when the system encounters a 0 in the field and attempts to divide by 0 in workbench calculations.

Chapter 2

Workbench Processes

This chapter tells you how the system works and explains background processing.

Introduction

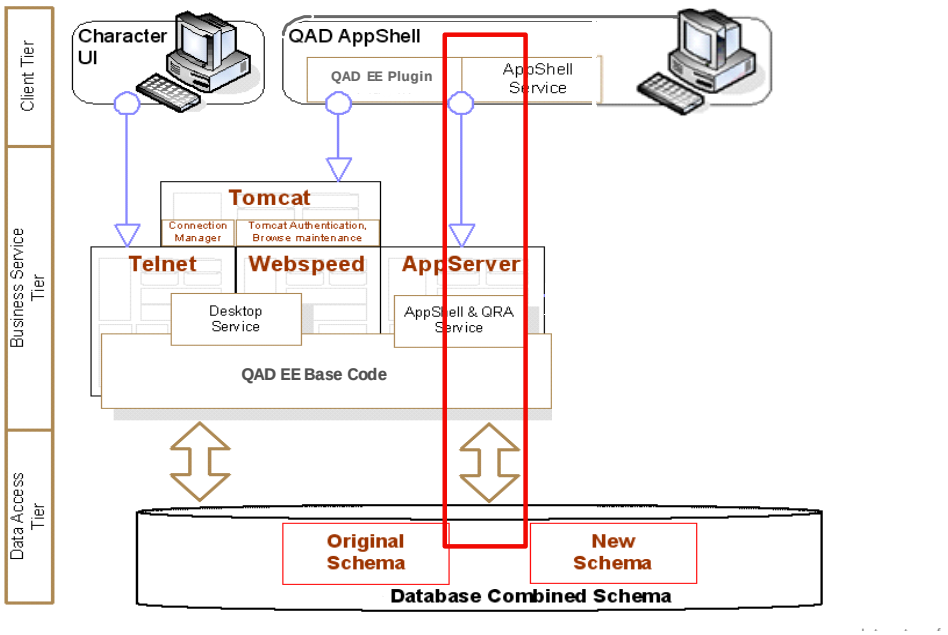
The Planning and Scheduling Workbenches, which includes two separate workbenches (MSW and PSW), a Shortage Report, several supporting browses and programs, and various frames that let you navigate resources, search, display capacity, and so on, is a plugin that runs inside the client; that is, it runs in the AppShell on the user's computer. The client connects with the database and business logic at the backend, or the *server*.

A *plugin* is a computer program that interacts with a host application (a Web browser or an e-mail client, for example) to provide a certain, usually very specific, function.

Schedulers create and maintain schedules in the workbenches, but they can also view how schedule changes impact the production line or work center load, supply and demand, and material availability. They also have simulation capabilities in the workbench. Typically, there is no impact on the database until the user commits by saving the data.

The following graphic depicts client-server interactions. The red line indicates the area of interest for MSW/PSW.

Fig. 2.1
Client-Server Interactions



Basic Processes

To understand what happens when users interact with the Planning and Scheduling Workbenches, the following topics present four common user processes:

- Search
- Modify
- Create new
- Save

These four processes typically constitute the basic processes performed in most applications. All of the processes, except modification, involve the client interacting with the server to access databases and run business logic.

Search Process

The following occurs when the user performs a search in the workbench:

1 The client sends search criteria to the program, running on the Progress appserver.

2 The program executes a search against the database.

3 The program transforms data into a format that the client needs.

4 The client receives data from the program and further transforms it.

5 The client loads data into memory, which, in turn, loads data on the screen.

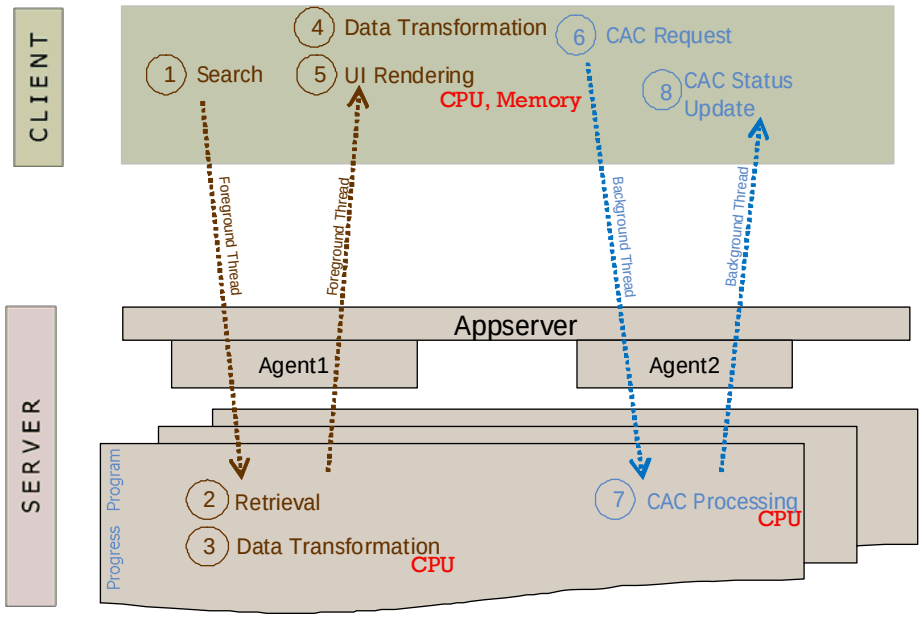
6 The client initiates a background thread to compute component availability check (CAC) statuses. The background thread calls the CAC computation program on the server.

Note A thread is a flow of control within a program, such as Java or C#, and is either a background or foreground thread. Background threads are similar to foreground threads except that they do not keep the managed execution environment running. So, when the last foreground thread stops, then all background threads stop, and the process stops.

7 The program computes statuses and returns them to the client.

8 The client loads statuses into memory, which, in turn, loads them on the screen.

Fig. 2.2
Search Process



Modify Process

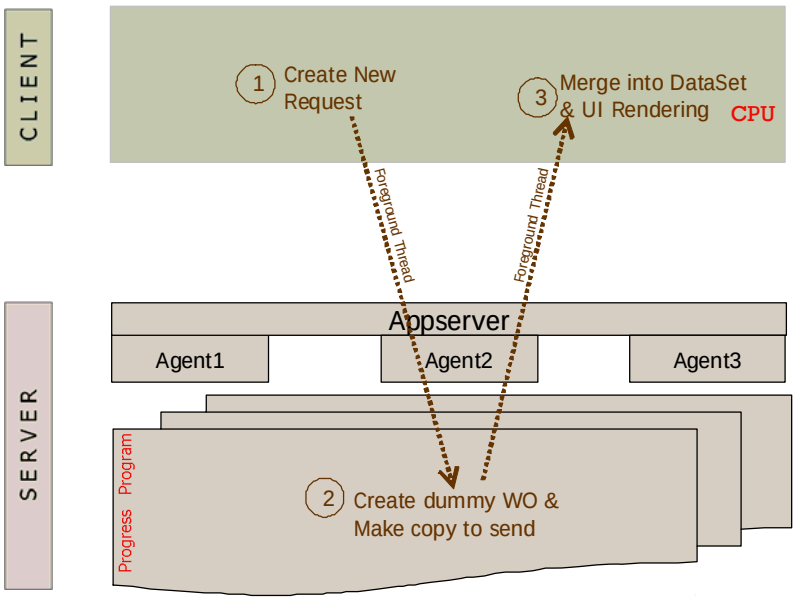
When the user modifies a production order, the client modifies data in memory. There is no client-server interaction, however.

Create New Process

When the user creates a new production order:

1. The client calls the program that runs on the Progress appserver.
2. The program makes a copy of the new production order, discards the production order created, but returns a copy.
3. The client loads the copy of the new production order into memory which, in turn, loads it on the screen.

Fig. 2.3
Create New Process



Save Process

When users save data in the workbenches, the following occurs:

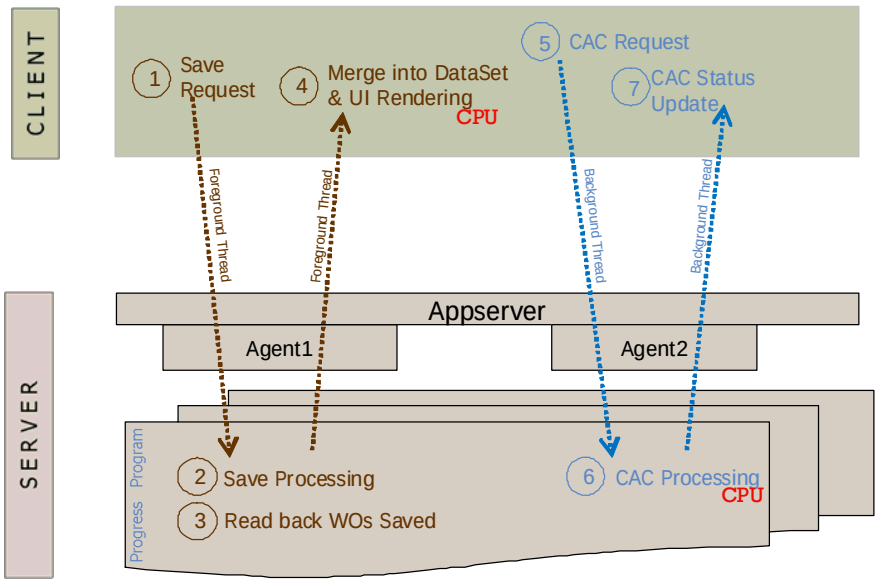
1. The client sends new/updated production orders to the program running on the Progress appserver.
2. The program saves the production orders after they have been validated.
3. The program reads back saved production orders.
4. The client reloads saved production orders into memory that it displays on the screen.
5. The client initiates a background thread to compute CAC statuses.

   The background thread calls the CAC computation program on the server.

6 The program computes statuses and returns the statuses to the client.

7 The client loads statuses into memory which, in turn, loads them on the screen.

Fig. 2.4
Save Process



As noted in the images, some steps demand significant CPU and/or memory resources. These steps largely decide the application response times. Chapter 4, “Configure,” discusses the parameters that affect system load. The key to getting better performance is to affect the system load and help these resource-intensive steps go faster.

Each request from the client results in the system running a Progress program on one of the appserver agents that is not busy. Until the processing for that request is complete, the agent is busy. Concurrent incoming requests are handled by one of the appserver agents that is available. In other words, concurrently running clients share a pool of appserver agents. It is important to ensure there are a sufficient number of agents in the pool to service all concurrent clients. Having insufficient agents results in incoming requests being queued, which adversely affects response times.

Steps that are resource intensive on the server are likely to keep the appserver agent busy longer. As the number of workbench users grows, it is important to monitor appserver agent usage and if necessary, to increase the agent pool size by configuring the appserver.

Chapter 3

System Load and Performance

This chapter describes system load and performance issues.

Introduction

The Planning and Scheduling Workbenches let schedulers simulate results. They can see the results of their updates and modifications before they commit this data to the database. The simulations on the schedule changes, however, can require that the system read and load a great deal of data from the database into system memory.

The system load comes from these elements:

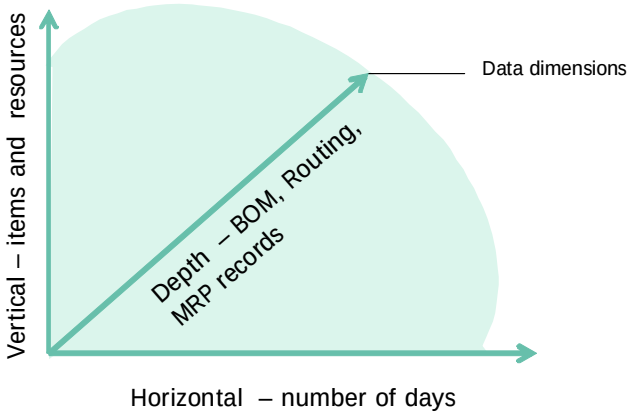
- Amount of data
- User preference parameters
- System configuration

System Load

System load comes in the following three dimensions:

- Item and resources
- Scheduling horizon days
- The number of MRP details; that is, production order records, component records (BOM), routing records, and so on

Fig. 3.1
System Load



Load on the client and server increases as any of dimensions increase, impacting system response time and, hence, performance.

Search criteria and user preference settings impact the three dimensions, too, as shown in the following table, which presents the performance impact by order of impact.

Table 3.1
Performance Impact by Order of Impact

Impact (1=Highest)	Area	Impact
1	Search criteria	This is the most significant impact, affecting the number of item/resources and therefore, indirectly the number of production orders, components, and routings.
2	Scheduling horizon	The horizon affects the number of production orders, components, and routings. After search criteria, this is the most significant parameter that affects performance.
3	Other user preferences	Setting other user preferences can impact performance; see “User Preferences Impact” on page 13.

User Preferences Impact

Use the data in Table 3.2 to determine whether a user preference can be disabled to improve performance. For example, if you disable CAC, you can improve performance when retrieving and saving data.

Table 3.2
User Preference Impact

Search/User Preference	Impact
Enable CAC	When enabled, the system computes component availability statuses during both search and save operations. This is a processor-intensive computation that can add a significant load on the server. ***Important*** If you do not need CAC functions, you should disable this feature as it can make a significant difference in performance.
Process Operation Details	Disabling this function prevents the search from transferring routing records to the client. Not having routing records on the client removes the required additional processing the system needs to keep routings updated. Users who schedule for production lines only should use this function.
Display Search Progress	When searching, this option makes the client divide the entire search into smaller groups. It also causes a progress bar to display as it processes each group. A processing overhead is associated with dividing the search into smaller groups. If the search takes too long, you should try disabling this feature.
CAC Horizon	The CAC horizon determines the number of days for which the system computes CAC status. This affects the number of MRP detail records that the server processes and sends to the client.

System Configuration

System performance is affected by system configuration, both hardware and software, as follows:

- Client PC

Processor speed and amount of memory have a very significant impact on response times. Chapter 2, “Workbench Processes,” discusses the common processes that demand significant memory and processor resources. Internal performance tests reveal that using PCs with older processors can result in less-than-optimal performance. So, it is important to have a client PC with a processor that has at least 3GB of RAM.

- Server machine

  Chapter 2, “Workbench Processes,” discusses a number of common processes that demand significant processor resources on the server machine. For this reason, it is important that the server machine has enough spare processing capacity to handle the load.

- Network

  All processes involving client-server interaction are affected by network speed; however, during data retrieval, the system transfers significant amounts of data from the server machine to the client over the network. System administrators should look for poor response times caused by network latency.

- Software configuration

  Progress administrators should ensure that there are a sufficient number of agents to service all concurrent QAD AppShell users; otherwise, the system can queue appserver requests from the client, resulting in poor response times.

Troubleshooting Performance

Although there is no single tool that lets you measure the application’s response times, system throughput, and system stress levels, you can improve performance by viewing system log files and curtailing the use of some user preferences to improve performance.

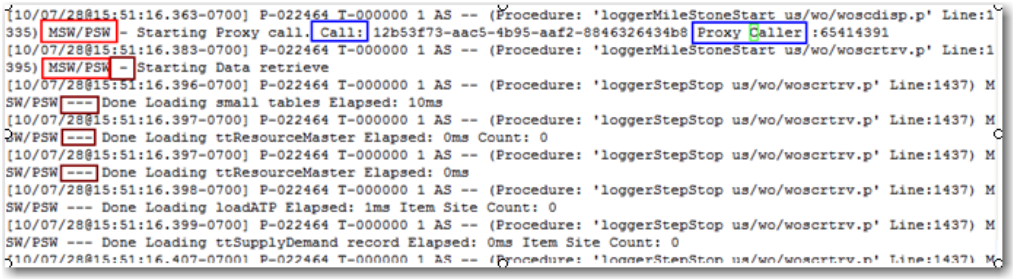
Log Files

You can examine the system log files to determine what took place in the system and how long it took. When users make requests to the back-end processor, or server, the system tracks the request in the appserver log file as well as in the client log file.

The client log file is on the user PC and only contains log messages pertaining to that user’s activity. The appserver log file, on the other hand, is shared by all users. So it contains log messages pertaining to activity of all users.

To effectively troubleshoot an issue, you should be able to associate a client request with a server response. Since the server log is for all clients, associating client requests with server response can be very difficult. To facilitate this association, both log files show Call ID and Proxy Caller ID. In addition, all messages are prefixed with the words MSW/PSW.

Fig. 3.2.
Appserver Log File, Proxy Caller



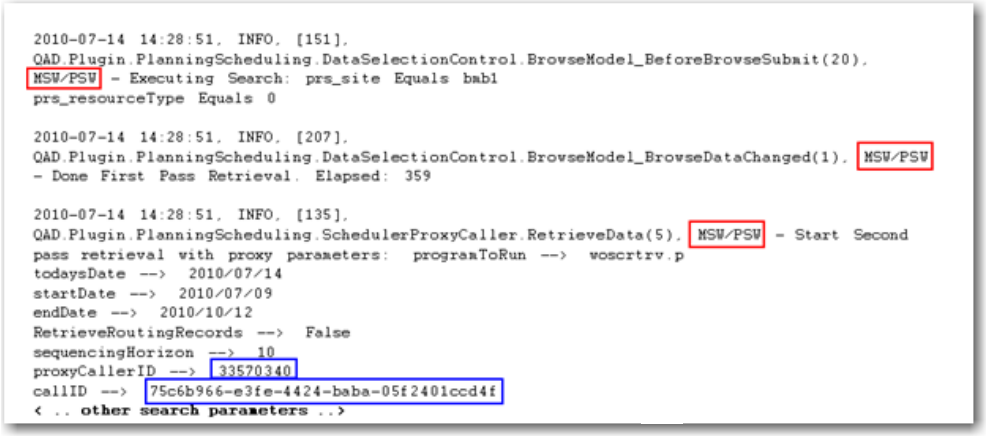
In the log file, you can determine when a user starts data retrieval as the log file depicts the start and finish of the data retrieval in milliseconds. The information can help you determine exactly how many records were retrieved, the type of record, how long the retrieval took, which user retrieved the records, and so on.

Knowing this can help you curtail long user retrievals. When retrieval issues arise, you can help users resolve the issues by successfully employing the MSW/PSW search filters or disabling functions through the user preferences.

You can find the client log file in .NET UI. At the top menu, select Help, then About, then View Log. If you operate on the appserver side, the system displays the appserver log file; if you operate on the client side, the system displays the client log file.

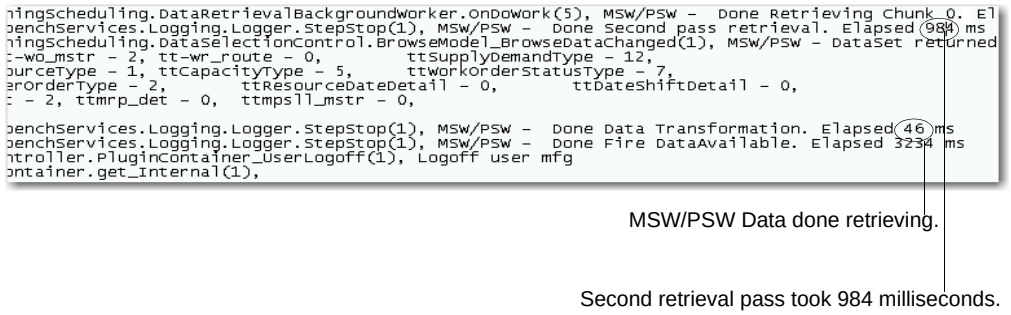
Each MSW or PSW request is preceded by the word MSW or PSW; see figure Figure 3.3.

Fig. 3.3.
Client Log File



The following graphic shows that the first pass of the data retrieval on the server took 259 milliseconds.

Fig. 3.4.
Appserver Log File, Timing



If you work on the client, the system retains a client log file that uses the same format and keeps track of the same events and requests that the appserver log file tracks.

Additional Files

The `/Application Data` folder of the client machine contains files for each plugin. Inside the `/plugin` folder is a layout folder that contains a list of XML files. The files are the views that Windows saves for the user as set in the View option. Note that the system saves views for Windows, not QAD EE users. That is, the system provides a capability that allows the layout to be saved to files; so, the XML files depend upon the view you have saved.

You can clear out all Application Data files for an AppShell installation; however, doing so would affect all data related to all plugins. So, if you choose to clear Application Data files, it clears all of previously mentioned files for MSW/PSW.

In addition, in the Application Data plugin folder, the following files exist:

- `Browsesearchconditions.xml`
- `Preferences.xml`

`Browsesearchconditions.xml` saves the last search that was performed by that Windows user. `Preferences.xml` contains the preferences saved by that Windows user.

Chapter 4

Configure

This chapter discusses the following configuration information:

Overview

This section describes how the plugin is internally configured. It describes the layout and location of the configuration XML file. It also provides an example and procedures to add a new browse to the workbench.

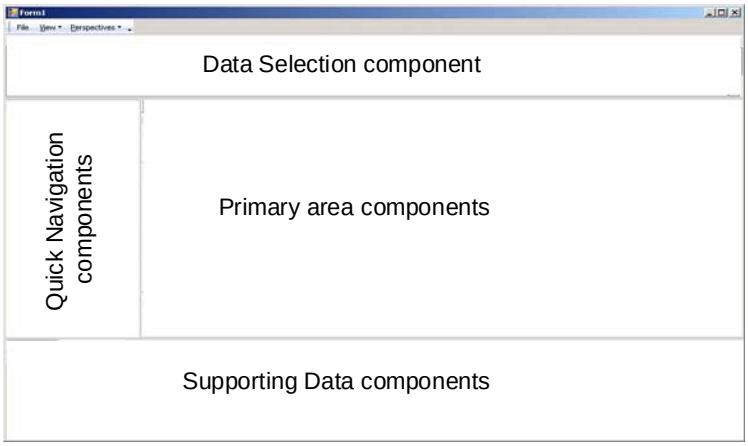
To understand how you configure the workbench, you must first understand the general layout of the workbench. The workbench user interface is divided into four areas:

1. Data selection: This area includes the basic search fields and expanded search fields that let you filter records the system retrieves in the workbenches. This area lets users select records from a very broad range—for example, all records associated with a site—to a very specific set of records—for example, a small set of orders for a particular production line. Users can optionally set user preferences so that the amount of time for the retrieval displays.
2. Primary data: This area is the heart of the MSW and PSW workbenches as well as the shortage report and CAC features. It is the area in which most users spend their time. Users select fields, rows, columns, or multiple fields, rows, and columns to manipulate data. They can also drag and drop data in this area within the MSW. Selections made in this area can change the data that displays in the supporting data area.
3. Quick navigation: This area is similar to the quick navigation that displays in many Internet browsers that helps you quickly navigate through an abundance of data.
4. Supporting data: This area includes browses and other programs that display data that supports the primary data. Programs that are new for the Planning and Scheduling Workbenches as well as browses are available through a row of tabs.

For the MSW/PSW plugin, the components that make up the four areas are as follows:

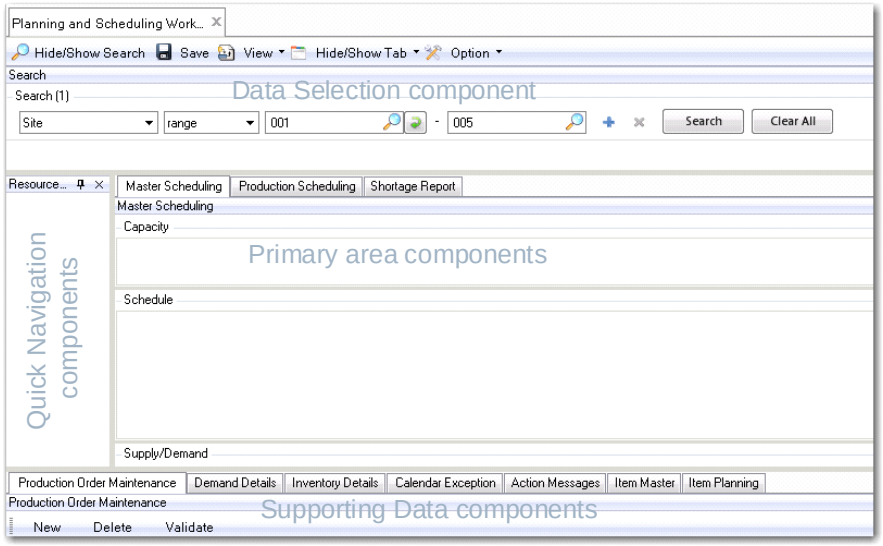
1. Browse component, `qpbr001.p`
2. Resource Navigator component
3. MSW, PSW and Shortage Report components
4. Browses Demand Details, Inventory Details, Calendar Exception, Action Messages, and Item Master data and maintenance programs, Production Order Maintenance, and Calendar Exception Maintenance

Fig. 4.1
Workbench Layout



The following graphic depicts the components on the actual Planning and Scheduling Workbenches UI to help relate the components to the interface.

Fig. 4.2
UI with Components



ControlConfig.XML File

The `ControlConfig.XML` file sets up and defines the QAD Planning and Workbench configuration. It defines the major frames that display and the actual fields that display within the supporting data frames; that is, the fields that display in the programs and browses that display in the tabs at the bottom of the workbench.

The XML consists of four tags, one for each area that the user interface has as shown in the blue boxes. Each tag can contain one or more control tags. Each control tag is for a component that displays in the user interface area. Each control tag consists of a class, assembly, and name tags. In addition, there may be other tags that are unique to a certain component. The visible tag is currently not used; see Figure 4.3.

Fig. 4.3
`ControlConfig.XML` Code

The file is located in the /`config` directory of your plugin installation on the local PC. In a typical system, the /`config` directory is located in a path like the following:

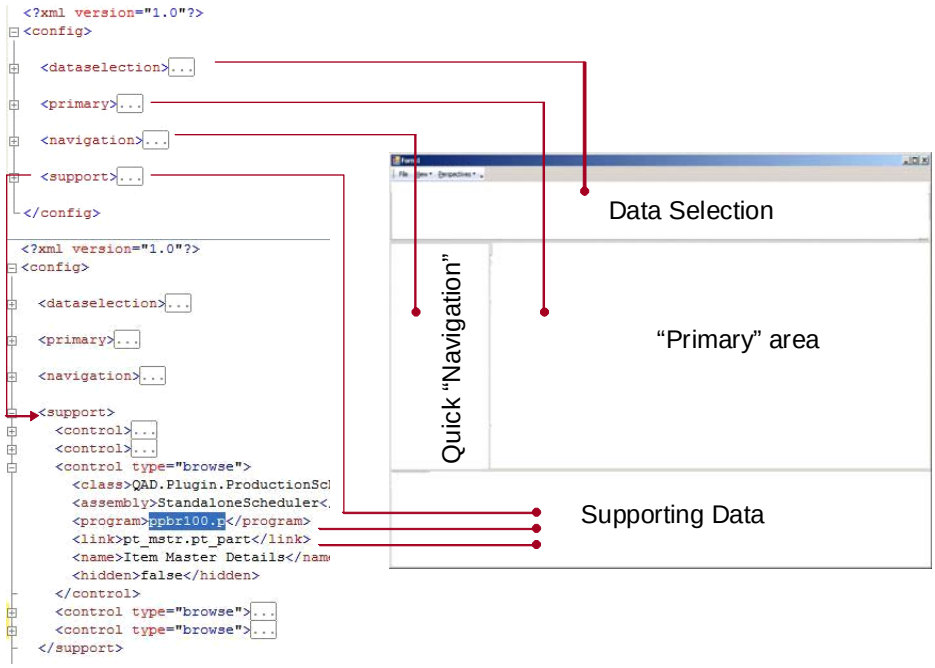
```
C:\Program Files\QAD\QAD Enterprise
Applications...\plugins\QAD.Plugin.PlanningScheduling\config\...
```

The file consists of control elements that define components of the workbench configuration in a hierarchical manner. For example, in the primary area are three controls are defined. These display on the user interface as three tabs:

- MSW
- PSW
- Shortage Report

Figure 4.4 depicts the code in the XML file and how it relates to the workbench framework.

Fig. 4.4
Code-UI Relationship



The Resource Navigator is defined as a control element for the `<navigation>` element, and Production Order Maintenance is defined as a `<control>` element for the `<support>` element. The actual programs are defined within the class `<class:>` definitions; see Figure 4.5.

Fig. 4.5
`ControlConfig.XML` File

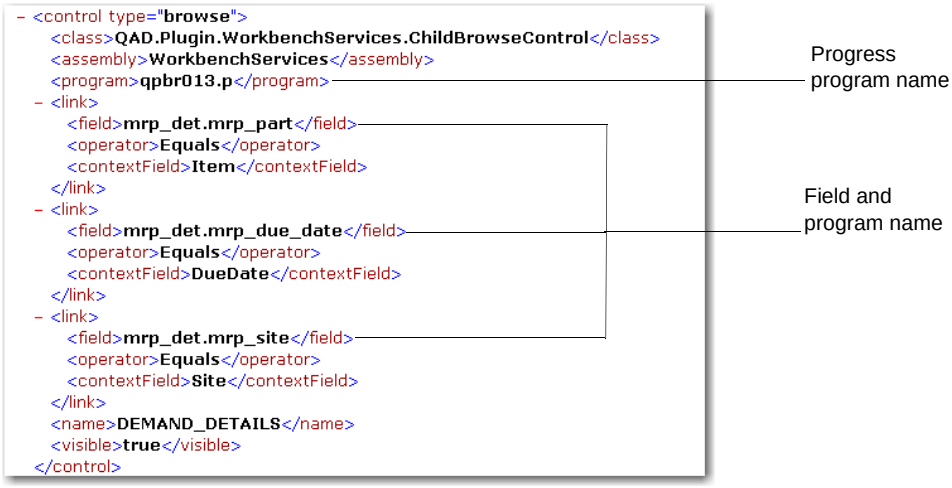
```
<?xml version="1.0" ?>
<config>
- <dataselection>
  - <control>
      <class>QAD.Plugin.PlanningScheduling.DataSelectionControl</class>
      <assembly>PlanningSchedulingPlugin</assembly>
      <visible>true</visible>
      <name>SEARCH</name>
    </control>
  </dataselection>
- <primary>
  - <control>
      <class>QAD.Plugin.PlanningScheduling.MasterSchedulerControl</class>
      <assembly>MasterScheduler</assembly>
      <visible>true</visible>
      <name>MASTER_SCHEDULING</name>
    </control>
  - <control>
      <class>QAD.Plugin.PlanningScheduling.ProductionSchedulerControl</class>
      <assembly>ProductionScheduler</assembly>
      <hidden>false</hidden>
      <name>PRODUCTION_SCHEDULING</name>
    </control>
  - <control>
      <class>QAD.WorkOrder.ShortageMonitorControl</class>
      <assembly>WorkOrder</assembly>
```

Defines the three major tabs that display in the workbench.

The following graphic depicts how the configuration, defined in `ControlConfig.XML`, relates to the UI.

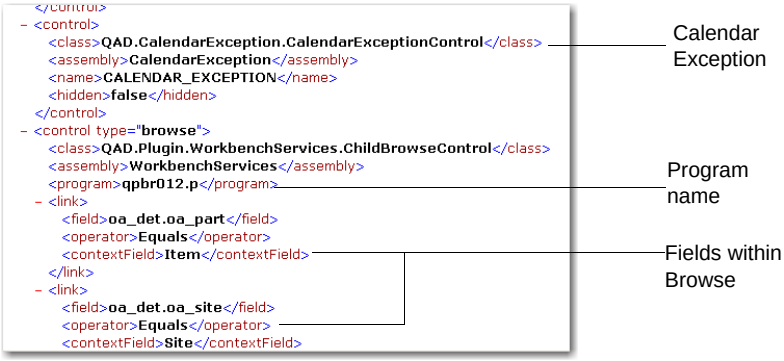
Browses that display within Production Order Maintenance are defined in `ControlConfig.XML` as `<control type=browse>`. The file defines the QAD EE Progress program name `(.p)`, then the fields within the browse are defined next, which includes the table and field name as defined within QAD EE.

Fig. 4.6
Browse and Field Definitions



Other programs in tabs that display at the bottom of the workbench are also defined as control elements. This is followed by defining the control type as browse, then field definitions that display within the browses. The following figure depicts the Calendar Exception tab.

Fig. 4.7
Other Program Definitions



Adding Browses to the Workbench

You can only configure the workbench to add browses. Adding new components other than browses to the XML requires that you have a knowledge of C# and workbench framework internals.

To configure, you start by accessing and editing the `ControlConfig.XML` file within the `/plugins` directory. In most cases, companies have their own data with which they work when creating master or production schedules, so the most common configuration change is the addition of a customer browse that pulls fields that hold data of interest to your schedulers.

To add a new browse that displays as a tab in the Supporting Data area alongside other QAD EE browses and programs, use the following example.

Example You have a QAD browse, *`YourBrowsebr001.p`*, customized for your business needs. You need the fields in the browse to display in a browse in the Planning and Scheduling Workbenches.

To easiest way to do this is to copy an existing browse configuration in the `ControlConfig.XML` file; then edit the copied text for your new browse entry.

To use this method, use the following procedure.

1 In `ControlConfig.XML`, locate and copy a browse control type definition:

```
- <control type="browse">
    <class>QAD.Plugin.WorkbenchServices.ChildBrowseControl</class>
    <assembly>WorkbenchServices</assembly>
    <program>qpbr011.p</program>
-   <link>
       <field>pt_mstr.pt_part</field>
       <operator>Equals</operator>
       <contextField>Item</contextField>
   </link>
-   <link>
       <field>si_mstr.si_site</field>
       <operator>Equals</operator>
       <contextField>Site</contextField>
   </link>
  <name>ITEM_PLANNING</name>
  <visible>true</visible>
  </control>
```

2 In the copied code, set the `<program>qpbr011.p</program>` entry to reflect the name of your new browse program, for example: `<program>`*`YourBrowsebr001.p`*`</program>`.

3 Determine which fields should be linked to the workbench, using QAD EE Browse Maintenance (36.20.13) to find valid field names for the fields that display in your browse; see Figure 4.8.

For example, if you are only interested in linking the item field with the workbench, you use Browse Maintenance for *`YourBrowsebr001.p`* to determine that the name of the item field is `xx_mstr.xx_part`.

4 Locate the `<link>` entry of the copied text, then enter the field name as depicted in Browse Maintenance within the `<field> </field>` parameters for the field to display in the browse.

Using the example in Step 3, enter:

```
<field>xx_mstr.xx_part</field>
```

5 Enter the operator as:

```
<operator>Equals</operator>
```

6 Enter the context for the field.

Using the example in Step 3, enter:

```
<contextField>Item</contextField>
```

For the Planning and Scheduling Workbenches, you can only enter a certain set of values for `<contextField>`. You cannot increase the set of values without QAD assistance. The set includes the following `<contextField>` values:

- Item
- Site
- DueDate
- Resource
- Resource2
- ResourceType

Where:

Resource is the production line ID or work center ID.

Resource2 is used for the machine.

ResourceType is zero (0) for the production line and one (1) for the work center or machine.

7 Continue editing existing `<link>` entries or add new `<link>` entries.

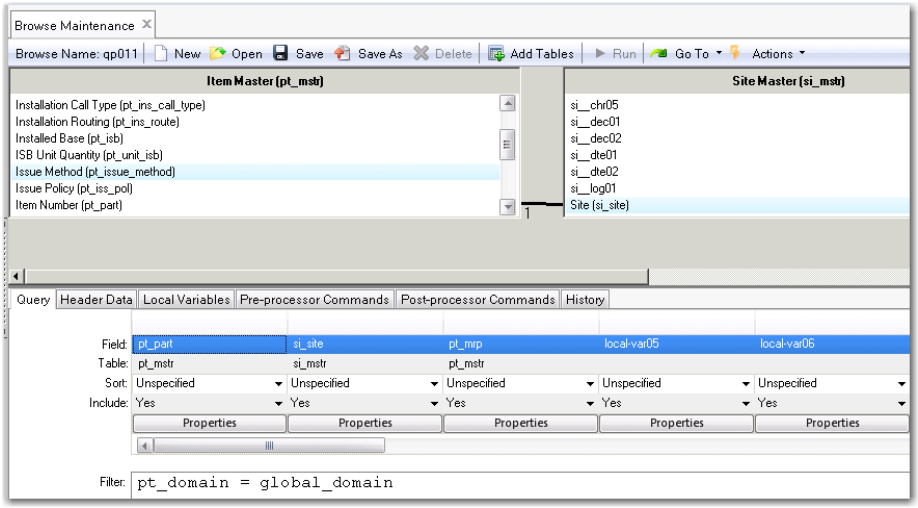
8 Name your browse and enter the name of the browse as:

```
<name>YourBrowse</name>
```

9 Save your entries.

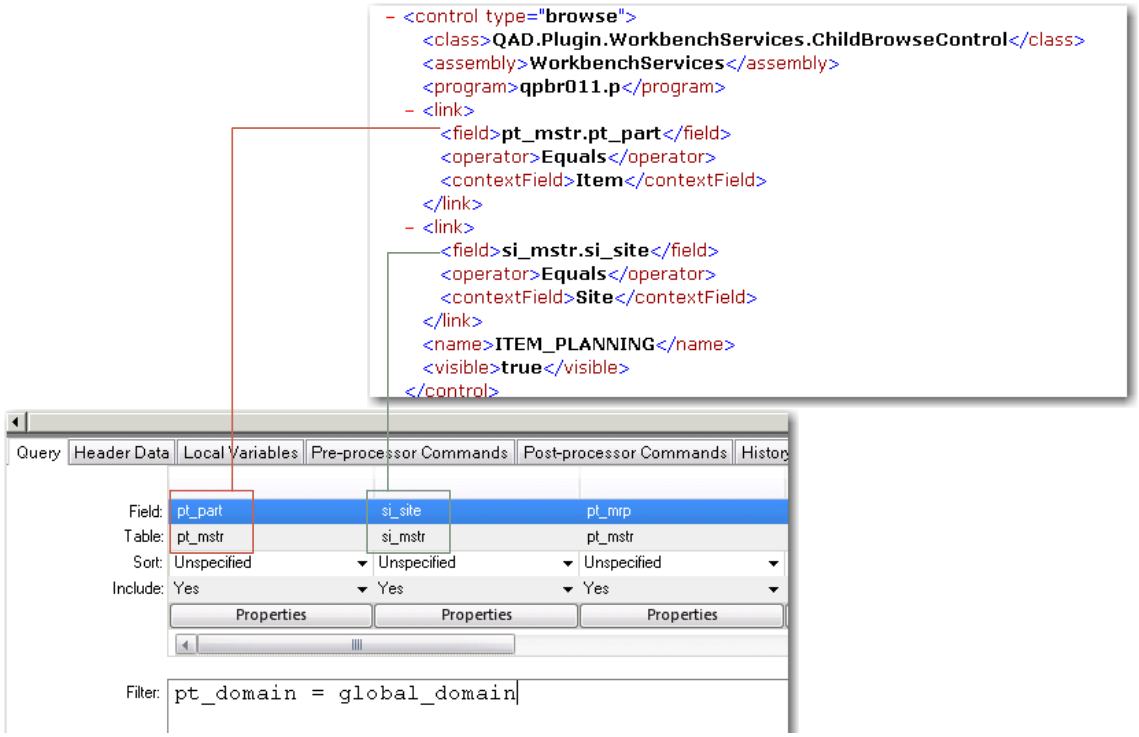
Important You should configure the XML for each client machine where the Planning and Scheduling Workbenches are configured.

Fig. 4.8
Browse Maintenance (36.20.13)



The following graphic depicts the code for a browse definition within `ControlConfig.XML`. It depicts how the fields names that you enter into the XML file were selected from Browse Maintenance.

Fig. 4.9
XML Fields



Saving Your Changes for New Releases

Should you update the MSW/PSW with additional releases, you must save your `ControlConfig.XML` file, then move it to another directory to hold while you install new files from the installation disk. Once the files install for the newer version of MSW/PSW, you can then move or copy your custom `ControlConfig.XML` file, overwriting the version of the file you installed for the update.