# Table of Contents

**Getting Started** ..................................................................................................................7  
Introduction ............................................................................................................................7  
Background Information ..........................................................................................................7  

**Windows: Navigating in the Operating System** .................................................................9  
Introduction ............................................................................................................................9  
Background Information ..........................................................................................................9  

**Admin Lab Exercise 1: Advantage Workflow Engine and Solution Environments** ........14  
Introduction ............................................................................................................................14  
Background Information ..........................................................................................................14  
Activity 1: Start the Engine through the AWESM Tool ..........................................................15  
Activity 2: Explore Virtual Terminal and HighJump One Platform User Interface (UI) [Part 1] .................................................................17  
Activity 3: Stop the Engine through the AWESM Tool ...........................................................27  
Activity 4: Explore Virtual Terminal and HighJump One Platform User Interface (UI) [Part 2] ....................................................................................................................................................30  
Activity 5: Stop / Start the Engine with the Operating System Services ................................34  
Activity 6: Stop and Start a Solution Environment ..................................................................37  
Activity 7: Stop the Engine with Active Users ........................................................................42  

**Admin Lab Exercise 2: Common Tasks in the Training Activities** .........................46  
Introduction ............................................................................................................................46  
Background Information ..........................................................................................................46  
Activity 1: Determine the Machine Name ...............................................................................47  
Activity 2: Run the Location Status Business Process ...........................................................49  
Activity 3: Determine the User ID and Password for an Application ..................................53  

**Admin Lab Exercise 3: User Profiles and Forklifts** .........................................................55  
Introduction ............................................................................................................................55  
Background Information ..........................................................................................................55  
Activity 1: Create a Material Handler User ............................................................................56  
Activity 2: Create a HighJump One Platform User .................................................................59  
Activity 3: Add a New Forklift ...............................................................................................68  
Activity 4: Change the Menu Level of a Material Handler User ............................................71  
Activity 5: Change the Menu Level of a HighJump One Platform User .............................75
Admin Lab Exercise 4: Terminal Menus
Introduction.........................................................................................................................81
Background Information......................................................................................................81
Activity 1: Explore the RF Menu Definitions in HighJump One Platform UI.................82
Activity 2: Modify the RF Menus.......................................................................................86

Admin Lab Exercise 5: Virtual Terminals and Web Terminals
Introduction....................................................................................................................98
Background Information.................................................................................................98
Activity 1: Define a Virtual Terminal in Advantage Commander.................................99
Activity 2: Define a Database Login for a Virtual Terminal.....................................107
Activity 3: Configure a Virtual Terminal on a Workstation........................................112
Activity 4: Define a Web Terminal in Advantage Commander.................................116
Activity 5: Define a Database Login for a Web Terminal............................................125
Activity 6: Configure a Web Terminal on a Workstation.............................................130
Activity 7: Configure an RF Terminal............................................................................134

Admin Lab Exercise 6: Advantage Platform Log
Introduction..................................................................................................................137
Background Information.................................................................................................137
Activity 1: View the Log Reports in Advantage Commander......................................139
Activity 2: Export the Log with the HighJump One Platform User Interface (UI) Export Feature..............................................................142
Activity 3: Export the Log with the Log Export Utility................................................145

Admin Lab Exercise 7: Status Consoles
Introduction..................................................................................................................148
Background Information.................................................................................................148
Activity 1: View a Log Message in a Status Console..................................................150
Activity 2: Change the Size of the Status Console Buffer........................................152
Activity 3: Copy Messages Out of the Status Consoles..............................................154

Admin Lab Exercise 8: Log Levels
Introduction..................................................................................................................157
Background Information.................................................................................................157
Activity 1: Elevate the Application Log Level with the AWESM Tool.......................159
Activity 2: Elevate the Application Log Level through the Control INI File.............163
Activity 3: Elevate the System Log Level with the AWESM Tool..............................167
Activity 4: Elevate the System Log Level through the Control INI File.....................171

Admin Lab Exercise 9: Managers
Introduction..................................................................................................................175
Background Information.................................................................................................175
Activity 1: Restart a Manager with the AWESM Tool................................................176

Admin Lab Exercise 10: Advantage Link
.................................................................
Admin Lab Exercise 15: Packages .........................................................306
Introduction ..................................................................................306
Background Information .................................................................306
Activity 1: Import a Package .............................................................308
Activity 2: Compile and Activate a Package ......................................312
Activity 3: Check In a Package .........................................................315
Activity 4: Rollback a Package .........................................................316

Admin Lab Exercise 16: Diagnostic Caching .................................319
Introduction ..................................................................................319
Background Information .................................................................319
Activity 1: Enable Diagnostic Caching ............................................320

Admin Lab Exercise 17: Other Administrative Tools ......................329
Introduction ..................................................................................329
Background Information .................................................................329
Activity 1: View the Log Entries in the Event Viewer ......................330
Activity 2: Export the Log Entries in the Event Viewer ....................332
Activity 3: Monitor Real-Time Counters with Performance Monitor ..333
Activity 4: Write Performance Monitor Counters to a Log .............336

Admin Lab Exercise 18: Schedule a Process Object .......................344
Introduction ..................................................................................344
Background Information .................................................................344
Activity 1: Technical Overview .......................................................345
Activity 2: Schedule a Process Object ............................................346

Admin Lab Exercise 19: SQL Server Profiler .................................352
Introduction ..................................................................................352
Background Information .................................................................352
Activity 1: Import an SQL Profiler Template ..................................353
Activity 2: Run an SQL Profiler Trace ..........................................355
Activity 3: Modify an SQL Profiler Trace ........................................358
Activity 4: Export an SQL Profiler Trace ........................................366

Admin Lab Exercise 20: Employee Messaging ...............................367
Introduction ..................................................................................367
Background Information .................................................................367
Activity 1: Send a Message to an Employee ....................................368
Activity 2: View a Response to an Employee Message ...................372
Getting Started

Introduction

This section consists of the following:

- How to use this manual
- Utilizing additional online lessons
- Getting help from an instructor
- Providing feedback

Background Information

How to use this Manual:

This manual contains self-directed exercises for the HighJump Software Administration VLab and instructor-led classes. Each exercise contains several hands-on activities. The exercises are designed to stand alone but the activities that make up the exercise tend to build upon each other.

Please print this manual and keep it for future reference. Each step in an activity is preceded by a check box. Many students find it helpful to check off each step as they progress in order to track their progress. If an exercise is not working, check your steps before contacting HighJump. Most errors are caused by a missed procedural step somewhere along the activity. It is recommended that the exercises be completed in the order they are presented.

Utilizing additional online lessons and help files:

Many exercises in this manual include online material references to assist your learning. The icon to the left precedes all such instruction and demonstration online materials. If you are taking an instructor-led class, you can bypass this material since it is part of the classroom experience. If you are taking a VLab class, then the online content provides instruction and demonstration for the exercise activity.

To access the online material, click one of the links under the Pre-Work section link in the panel on the right side of VLab environment.
Getting help from an instructor

If you need to contact an instructor, click the Ask an Instructor button in the panel on the right side of VLab environment.

Providing feedback

When you complete the course, please fill out the online evaluation. It is completely anonymous and will help HJU improve its training deliverables. You can access the online evaluation by clicking on the Evaluate HJU VLabs button in the panel on the right side of the VLab environment.
Windows: Navigating in the Operating System

Introduction

This exercise consists of the following activities:

1. Opening an Application
2. Exiting the Start Window and the Apps Window

Background Information

This class utilizes the Windows Server 2012 R2 operating system. This operating system is not as intuitive as some of the previous versions. Due to some of the fundamental changes introduced by Microsoft, this exercise demonstrates some of the basic operations within the operating system. Additionally, it introduces the language used in this manual to describe these basic operations.
Activity 1: Opening an Application

**Procedures:** The procedures below demonstrate how to open an application in the operating system.

- Look in the lower left corner of the main window.

The traditional “Start” button that appeared in previous versions of the operating system has been replaced by the modern Windows icon. Although the icons are different between the traditional version and the newer one, both buttons function in a similar manner.

Even though the “Start” text no longer appears on this button, this manual will refer to it as the “Start” button.

- Click the **Start** button.

The traditional "Start" menu that appeared in previous versions of the operating system has been replaced by a modern, paneled look. While this window has some similar features to the traditional “Start” menu, it has some added capabilities as well.

- Look in the lower left corner of the Start window.
There is an arrow icon inside of a white circle. This button functions similarly to the traditional “All Programs” menu under the Start menu in previous versions.

Even though the “All Programs” text no longer appears on this button, this manual will refer to it as the “All Programs” button.

☐ Click the All Programs button.

The system displays a list of all applications installed on the machine in the Apps window. This list is comparable to the applications that appeared under the All Programs menu in previous versions. The options in white are the application names. The text that appears in blue are the grouping names which is comparable to the folder names in previous versions.

When an instruction in this manual directs you to open a new application, it will include both the grouping name and the application name. The instruction will look similar to “Click the Administrative Tools | Task Scheduler menu option”

☐ Use the horizontal scroll bar to scroll all the way to the right.

☐ Click the Windows Accessories | Calculator menu option.

The system closes the Apps windows and launches the calculator application.
Activity 2: Exiting the Start Window and the Apps Window

Procedures: The procedures below demonstrate how to return to the desktop if you are in the Start Window or the Apps Window.

The desktop is where you can view the applications that are running. In this example you can see the calculator application running on the desktop.

☐ Click the Start button.

The system displays the paneled Start window.

☐ Click the Desktop panel that appears immediately below the Administrative Tools panel and the File Explorer panel.
The system returns to the desktop and displays the calculator application.

Next you will navigate to the Apps window and then return to the desktop.

☐ Click the **Start** | **All Programs** button.

The system displays the Apps window.

While there are a couple ways of returning to the desktop from the Apps window, the easiest method is to leverage the escape key.

☐ Press the **Escape** key.

The system returns to the desktop and displays the calculator application.

☐ Close the **Calculator** application.
Admin Lab Exercise 1:
Advantage Workflow Engine and Solution Environments

Introduction

This exercise consists of the following activities:
1. Start the Engine through the AWESM Tool
2. Explore the Virtual Terminal and HighJump One Platform User Interface [Part 1]
3. Stop the Engine through the AWESM Tool
4. Explore the Virtual Terminal and HighJump One Platform User Interface [Part 2]
5. Stop / Start the Engine with the Operating System Services
6. Stop and Start a Solution Environment

Background Information

The Advantage Workflow Engine is a collection of executables (called managers) that run on the Application Server. It is also simply referred to as the engine. It is a key component of the Advantage Platform because it controls the execution of the Advantage system. In particular, it executes the Advantage application and database interactions and controls the attached devices, such as readers and printers. The Advantage Workflow Engine runs as an operating system service.

The Advantage Workflow Engine is similar to the engine in an automobile. If the car engine is running, you can travel to other destinations. If the car engine is not running, then you can only sit in the garage and listen to the radio. Similarly, if the Advantage Workflow Engine is running, then the HighJump system is running and you can execute the full suite of business processes like receiving, picking, packing, and shipping. However, if the Advantage Workflow Engine is not running, then the HighJump system is down, and you can only perform a limited number of activities.

Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Workflow Engine menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Start the Engine through the AWESM Tool

Background: The system provides two methods for starting the engine. The first method uses a tool called the Advantage Workflow Engine Service Manager. It is commonly referred to as the AWESM (sounds like "awesome") tool. The second method will be discussed in a later activity. While the AWESM tool has other capabilities, its primary purpose is to start and stop the engine.

Scenario: The HighJump implementation team has installed the software on your development servers. Your QA team has asked you to start the HighJump system so that they can familiarize themselves with several of the base application business processes.

Procedures: The procedures below demonstrate how to start the Advantage Workflow Engine through the AWESM tool. In essence, this starts the HighJump system.

☐ Choose the Start | All Programs | HighJump Software | Advantage Workflow Engine Service Manager menu.

The system displays the engine icon in the system tray. The color of the icon is significant. A red icon indicates that the Advantage Workflow Engine is not running.

☐ Double-click the engine icon in the system tray.

The system opens the Advantage Workflow Engine Service Manager window. The color of the engine icon in the top portion of the window should match the color of the icon in the tool tray. The red icon indicates that the Advantage Workflow Engine is not running.
Click the **Start** button on the top portion of the window.

Several changes take place when you click the Start button.

- The engine icon in the top portion of the window changes from red to yellow to green.
- The engine icon in the tool tray changes from red to yellow to green.
- The Start button is disabled and the Stop button is enabled in the top portion of the window.
- The system displays several solution environments in the lower portion of the window each with their own individual engine icons.

When the engine icon in the top portion of the screen is green, and all of the solution environment icons are green, then the HighJump system is fully running. At this point the QA team could begin their analysis of any of the business processes in the HighJump system.
Activity 2: Explore Virtual Terminal and HighJump One Platform User Interface (UI) [Part 1]

Background: The Virtual Terminal is a program that looks and behaves like a physical RF reader. All of the business processes that run on a physical reader can also run on a Virtual Terminal. The HighJump One Platform UI is a set of web pages that allows you to view and modify HighJump data in the database tables. Both of these programs are critical components of the overall HighJump system.

Scenario: The QA team has asked you to start the HighJump system so that they can familiarize themselves with several of the base application business processes. You started the engine, and now you want to spot check a couple components of the HighJump system before you tell them that the system is running.

Procedures: The procedures below demonstrate how to spot check the Virtual Terminal and the HighJump One Platform UI components. Because there are some interdependencies between these components, the procedures often switch between the two programs. Additionally, you will explore a handful of other web pages that go outside the scope of the spot check scenario. These additional web pages will serve as a basis of understanding for a later activity.

Choose the Start | All Programs | HighJump Software | Advantage Virtual Terminal menu.

The system opens the Virtual Terminal program.

The Virtual Terminal displays a welcome screen and it prompts for a user id. The welcome screen indicates that the Virtual Terminal is communicating with the Advantage Workflow Engine. However, it does not indicate that the Virtual Terminal is fully functional. In order to fully test the Virtual Terminal and its connectivity to the engine and the database, you should log into it with a user id, a password, and a forklift. You can find a valid user id by navigating to a web page within the HighJump One Platform UI.

Open Internet Explorer
☐ Enter the following URL in the address bar.

http://[web server]:30000/core/default.html

The system displays the login screen.

☐ Enter Administrator in the User Name edit box.

☐ Enter HJSPASS in the Password edit box.

☐ Click the Login button.

The system opens the HighJump One home page and displays the logo in the middle.

This home page is the launching pad from which you can view and modify the warehousing data in your Supply Chain Advantage (SCA) system. The Menu Bar on the left side consists of three icons. The Application Menu is where the various applications and their associated pages reside. Spaces are distinct work area containers that can be accessed concurrently. And Favorites are bookmarks of frequently visited pages.
☐ Click the **Application Menu | Supply Chain Advantage | Warehouse Advantage | Employees | Employees** menu.

The system displays the Search Employees page.

☐ Click the **Query** button in the **Action Bar**.

The system displays a list of all employees for the HighJump system.
The employees listed on this page all have a profile that can be used to access the Virtual Terminals (…and Web Terminals and RF terminals.) The value in the Employee ID column of the web page corresponds with the USER ID prompt on the Virtual Terminal.

You could use any of the employees listed on the web page to log into the Virtual Terminal. However, some of the employees do not have access to all of the business processes in the Virtual Terminal. Amy is an administrator, and therefore she has complete access to all of the business processes. This activity and several that follow use Amy's profile to log into the Virtual Terminal.

☐ In the Virtual Terminal, type **AMY** at the USER ID prompt.

The system advances to the password prompt.
Amy’s password can be also be found by navigating through the HighJump One Platform UI pages. However, it is not on the page which lists the employees in a grid format.

☐ In the HighJump One Platform UI, click the **number** to the left of Amy’s name.

The system displays a page containing all of the attributes associated with Amy. One of these attributes is her password.
The value of the Password edit box corresponds with the PASSWORD prompt on the Virtual Terminal. In the sample data delivered with the system, Amy’s password is the same as her user name. However, this is only true in the sample data. In your production system this will most likely not be the case.

In the Virtual Terminal, enter AMY at the PASSWORD prompt.

The system advances to the equipment / zone prompt.

For most customers, the value entered at this prompt is a forklift. Anytime inventory is moved from a location onto the user’s physical forklift, the HighJump system shows that the electronic inventory is associated with the forklift. In the HighJump system, the forklift is basically a location on wheels. The forklifts, and other locations, are located within a different set of web pages.
In the HighJump One Platform UI, click the Menu | Supply Chain Advantage | Warehouse Advantage | Warehouse Setup | Locations menu.

The system displays the Search Locations page.

- Choose Warehouse 01 in the Warehouse ID drop-down list.
- Choose Forklift in the Location Type drop-down list.
- Click the Query button in the Action Bar.

Because of the information provided on the previous web page, the system displays a list of all forklifts in Warehouse 01.

The forklifts listed on this page can be used as part of the login process for the Virtual Terminals (and Web Terminals and RF terminals.) The value in the Location ID column of the web page corresponds with the EQUIPMENT / ZONE prompt on the Virtual Terminal. As you review the locations listed on the page, you will see that all of the locations begin with “FA” followed by the employee id of an employee. This is a convention that is used in the sample data only. The data in your production system will most likely not use this convention. While you could use any of the forklifts listed on this page for this activity, the FAAMY forklift has been specifically created to use in conjunction with Amy’s login.
In the Virtual Terminal, enter FAAMY at the EQUIPMENT \ ZONE prompt.

The system advances to the main menu prompt. Because AMY is an administrator, the system displays the menu structure for the complete set of business processes.

At this point, the spot check is complete. You have validated that you can log into a Virtual Terminal. And you have validated that you can navigate through several HighJump Platform One UI pages. You can be confident that the primary components of the HighJump system are working correctly, and you can communicate that to the QA team.

Even though you have completed the spot check, this activity includes one additional set of steps. These steps are exploratory in nature and not necessarily related to the spot check. While the web pages you review below will look similar to the previous ones you visited, a later activity will reveal that one of them is fundamentally different from the others. These additional web pages revolve around the concept of
replenishments. Understanding the concept of a replenishment is not necessary for the purpose of this activity.

☐ In the HighJump One Platform UI, click the Menu | Supply Chain Advantage | Advantage Dashboard | Planned Moves | Replenishments | Generate Top Off Replenishments menu.

The system displays the Generate Top Off Replenishments page.

☐ Choose Warehouse 01 in the Warehouse ID drop-down list.

☐ Enter P114 in the Starting Location edit box.

☐ Enter P116 in the Ending Location edit box.

☐ Click the Query button in the Action Bar.

The system displays a list of locations that are eligible for replenishment. Each eligible location has a Generate Replenishment link on the right side of the grid.
☐ Click the **Generate Replenishment** link on the right end of the **P115** record.

Behind the scenes, the system generates a replenishment for the P115 location, and then it returns to the page which displays the eligible locations. The P115 record is notably absent from the page.

Keep in mind, that you have performed all of the steps in this activity while the Advantage Workflow Engine is running. In a future activity, you will see some different results when you will attempt to perform these same steps while the Advantage Workflow Engine is down.
Activity 3: Stop the Engine through the AWESM Tool

Background: The system provides two methods for stopping the engine. The first method uses the Advantage Workflow Engine Service Manager (AWESM). The second method will be discussed in a later activity.

Stopping the Advantage Workflow Engine has a significant impact on the users of the HighJump system. When the engine is down, the users cannot use any of the terminal devices (RF, Virtual or Web.) And even some of web pages will not function. Additionally, if you shut the engine down while users are in the middle of business processes, you will most likely have some data that needs to be cleaned up. The data clean up that results from an improperly shut down engine can be a painful task. Shutting down the engine of a production system is not something that should be treated lightly.

Scenario: You contracted with the HighJump Professional Services Team to make several changes to the receiving business process. They have delivered the changes to you, and you have tested them in the testing environment. Now you want to promote the changes to the production environment. Part of the promotion process includes shutting down the engine. You want to promote the changes during the lunch break, so you have notified all users to log off the readers before they take their lunch break.

Procedures: The procedures below demonstrate how to stop the Advantage Workflow Engine through the AWESM tool. Additionally, they demonstrate how to monitor a couple items that should be checked prior to shutting down the engine.

As mentioned above, if you shut the engine down while users are in the middle of a business process, you may create a situation that requires some data cleanup. The best practice is to ensure that all users have logged off the readers before you shut down the engine.

☐ Open the Advantage Workflow Engine Service Manager, if necessary.

The system indicates that there is one active terminal and one active session in the WA solution environment. A terminal corresponds with an active terminal license. A session corresponds with an active session in the solution environment.

<table>
<thead>
<tr>
<th>Solution Environment(s)</th>
<th>Name</th>
<th>State</th>
<th>Terminals</th>
<th>Sessions</th>
<th>Version</th>
<th>Activation Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADVANTAGEPLATFORM</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2016-07-20 16:24:19</td>
</tr>
<tr>
<td></td>
<td>ADV/LINK/FILE</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2016-08-08 15:58:51</td>
</tr>
<tr>
<td></td>
<td>AFA</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2016-05-18 15:13:46</td>
</tr>
<tr>
<td></td>
<td>WA</td>
<td>Running</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>2016-05-17 17:05:40</td>
</tr>
<tr>
<td></td>
<td>WA/PROCESSORS</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2016-06-08 15:05:17</td>
</tr>
</tbody>
</table>

Because you just logged into the virtual terminal as AMY, you know that the one session logged into the WA solution environment belongs to AMY. However, in a production environment you may not know who is actively logged into the terminals. The procedures below demonstrate how to obtain additional information about the users that are logged into the WA solution environment.
☐ Click the **WA solution environment** in order to highlight it.

☐ Click the **Sessions** button.

The system opens another window which displays all of the active sessions for the WA solution environment. This window shows terminals (RADIO_01), and it shows other “system-type” sessions. These “system-type” sessions are essentially sessions running in the background.

When you want to shut down the engine, you are most likely primarily concerned with the terminals. The next instruction demonstrates how to filter this list so that it only displays information related to terminals.

☐ Choose the **Show Active Devices** in the **View** drop-down list.

The system filters the above list, and only displays active terminals.

The system indicates that there is one active terminal (RADIO_01), and that AMY is the one who is logged into the terminal. The system will allow you to shut down a solution environment with active users, and you will learn how to do that in a later activity. In this instance, you will simply contact AMY, and have her log out of the terminal.

☐ Click the **X** in the ***upper right corner*** of the **Sessions** window.

☐ Press the **F1 key** on the Virtual Terminal.

The virtual terminal returns to the USER ID dialog. Additionally, the AWESM tool indicates that there are no active sessions in the WA solution environment.
At this point, you can shut down the entire system without impacting any end users.

☐ In the AWESM tool, click the Stop button on the top portion of the window.

☐ Click the Yes button.

Several changes take place when you click the Yes button.

- The engine icon in the top portion of the window changes from green to yellow to red.
- The engine icon in the tool tray changes from green to yellow to red.
- The Start button is enabled and the Stop button is disabled in the top portion of the window.
- The system removes all of the solution environments from the lower portion of the window.

The engine is down when the engine icon turns red on the top portion of the Advantage Workflow Engine Service Manager window. As mentioned above, some aspects of the HighJump product suite will not work when the engine is down, while others will work correctly. The next activity explores how the various software components are effected by an engine-down situation.
Activity 4: Explore Virtual Terminal and HighJump One Platform User Interface (UI) [Part 2]

Background: In a previous activity you explored the Virtual Terminal and several HighJump One Platform UI pages while the engine was running. In this activity you will revisit the Virtual Terminal and some additional HighJump One Platform UI pages while the engine is down.

Scenario: You contracted with the HighJump Professional Services Team to make several changes to the receiving business process. They have delivered the changes to you, and you have tested them in the testing environment. Now you want to promote the changes to the production environment. Part of the promotion process includes shutting down the engine. This promotion will take place during your company’s busy season. The workers in the warehouse have a significant workload and they are reluctant to stop working for any period of time during the promotion. They ask you which components will work and which ones will not work while the promotion is in progress.

Procedures: The procedures below demonstrate which components run and which components do not run while the engine is down.

- Shut down the engine, if necessary.
- Open the Virtual Terminal, if necessary.

The main screen of the Virtual Terminal indicates that the engine is not running. The system does not allow you to log into the Virtual Terminal.

The key point is that if the engine is not running, then you cannot run any business processes from a Virtual Terminal. The same holds true for the RF terminals and the Web Terminals. No activity will take place on any terminal as long as the engine is down.
☐ Open the **HighJump One Platform UI**, if necessary.

☐ Click the **Menu | Supply Chain Advantage | Warehouse Advantage | Employees | Employees** menu.

☐ Click the **Query** button in the **Action Bar**.

Even though the engine is not running, the HighJump One Platform UI allows you to navigate through the various pages to reach the employee information.

<table>
<thead>
<tr>
<th>#</th>
<th>Employee Name</th>
<th>Employee ID</th>
<th>Warehouse ID</th>
<th>Menu Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AMY ADMINISTRATOR</td>
<td>AMY</td>
<td>01</td>
<td>BASE</td>
</tr>
<tr>
<td>2</td>
<td>CURT CUSTOMER</td>
<td>CURT</td>
<td>01</td>
<td>BASE</td>
</tr>
<tr>
<td>3</td>
<td>ED EXECUTIVE</td>
<td>ED</td>
<td>01</td>
<td>BASE</td>
</tr>
</tbody>
</table>

☐ Click the **Supply Chain Advantage | Warehouse Advantage | Warehouse Setup | Transportation | Carriers** menu.

☐ Click the **Query** button in the **Action Bar**.

The system displays a list of all carriers.

<table>
<thead>
<tr>
<th>#</th>
<th>Carrier Name</th>
<th>Carrier Code</th>
<th>SCAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Airborne</td>
<td>AIR</td>
<td>AIRB</td>
</tr>
<tr>
<td>2</td>
<td>Company Fleet</td>
<td>FLEET</td>
<td>FLT</td>
</tr>
<tr>
<td>3</td>
<td>DHL</td>
<td>DHL</td>
<td>DHL</td>
</tr>
</tbody>
</table>

☐ Click **any number** to the left of any of the carriers.

The system displays all of the attributes associated with the given carrier.
The systems behavior here is similar to the employee web pages – you can navigate through the various carrier web pages even though the engine is down. The key point is that even when the engine is not running, you can perform several tasks within the HighJump One Platform UI. However, not all pages work when the engine is down. The next steps demonstrate a page that will not work while the engine is down.

☐ Click the Menu | Supply Chain Advantage | Advantage Dashboard | Planned Moves | Replenishments | Generate Top Off Replenishments menu.

The system displays the page which captures the boundaries for the replenishment locations.

☐ Choose Warehouse 01 in the Warehouse ID drop-down list.

☐ Enter P110 in the Starting Location edit box.

☐ Enter P113 in the Ending Location edit box.

☐ Click the Query button in the Action Bar.
The system displays a list of locations that are eligible for replenishment.

<table>
<thead>
<tr>
<th></th>
<th>Warehouse ID</th>
<th>Item Number</th>
<th>Location</th>
<th>UOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01</td>
<td>NNFG00000</td>
<td>P111</td>
<td>EA</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>NNFG07505</td>
<td>P112</td>
<td>EA</td>
</tr>
<tr>
<td>3</td>
<td>01</td>
<td>NNFG50680</td>
<td>P113</td>
<td>EA</td>
</tr>
</tbody>
</table>

☐ Click any **Generate Replenishment** link on the far right of the record list.

After a brief delay, the system displays an error message. The error message is somewhat cryptic, but it means that the engine is not running. This particular page will not run correctly unless the engine is running.

☐ Click the **Dismiss** button.

In summary, you must consider the impact on all components of the HighJump system when you stop the Advantage Workflow Engine. Business processes will not run on a terminal devices (Virtual, Web, or RF) when the engine is shut down. In contrast, some HighJump One Platform UI pages work correctly and others generate errors when the engine is shut down. (The distinction between which pages work and which pages do not work under this scenario goes beyond the scope of this class. If you are interested in knowing exactly which of your pages will not work when the engine is down, then test your web pages in your development environment.)
Activity 5: Stop / Start the Engine with the Operating System Services

**Background:** The system provides two methods for stopping and starting the engine. In a previous activity, you learned how to stop the engine through the Advantage Workflow Engine Service Manager (AWESM). The system also provides a method for stopping (and starting) the engine through the services component of the operating system. The engine runs as a service behind the scenes, so it can be controlled as a service.

If you want to stop or start the engine (all solution environments), then the two options are virtually equivalent. Which option you choose is a matter of preference and not functionality. In a future activity, you will learn how to stop an individual solution environment. In that case the AWESM tool provides a significant advantage.

**Scenario:** You contracted with the HighJump Professional Services Team to make several changes to the receiving business process. They have delivered the changes to you, and you have tested them in the testing environment. Now you want to promote the changes to the production environment. Part of the promotion process includes shutting down the engine, and then starting it after you have promoted the changes.

**Procedures:** The procedures below demonstrate how to stop and start the Advantage Workflow Engine through the services component of the operating system.

When you stopped the engine through the AWESM tool, you validated that there were no RF users (or Virtual Terminal users or Web Terminal users) on the system. You should perform the same checks when you shut down the engine through the operating system services. These procedures do not walk through those preliminary checks.

- Start the engine through the AWESM tool, if necessary.
- Open the **Virtual Terminal**, if necessary.
- Choose the **Start | All Programs | Administrative Tools | Services** menu.

The system displays a list of all operating system services. The service in question is the one titled Advantage Workflow Engine Service. The Status column reads “Running.” This indicates that the HighJump engine is running.

- Right-click the **Advantage Workflow Engine Service** option.
- Choose the **Stop** menu.
After a couple of minutes the system changes the Status of the service to an empty value. This indicates that the engine is not running.

You can validate that the engine is not running by looking at the color of the icon in the top portion of the AWESM tool, as well as by looking at the main screen of the Virtual Terminal.

- Right-click the **Advantage Workflow Engine Service** option.
- Choose the **Start** menu.

After a couple of minutes the system changes the Status of the service to “Running”. This indicates that the engine is running.

You can validate that the engine is running by looking at the color of the icon in the top portion of the AWESM tool, as well as by looking at the main screen of the Virtual Terminal.
Admin Lab Exercise 1:
Advantage Workflow Engine and Solution Environments
Activity 6: Stop and Start a Solution Environment

**Background:** Definition: A solution environment is a collection of devices, global parameters, and business rules (workflows) designed to solve a specific supply chain problem.

The bottom portion of the AWESM window displays the solution environments.

<table>
<thead>
<tr>
<th>Solution Environment Name</th>
<th>Long Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADVANTAGEPLATFORM</td>
<td>Advantage Platform</td>
<td>Manages messages sent between solution environments. There is no user interface to this solution environment.</td>
</tr>
<tr>
<td>ADVLINKFLATFILE</td>
<td>Advantage Link for Flat File</td>
<td>Manages data exchanged between HighJump applications and the host using text files.</td>
</tr>
<tr>
<td>AFA</td>
<td>Advantage Fulfillment Application</td>
<td>Manages the release of waves and loads to the floor for picking.</td>
</tr>
<tr>
<td>WAPROCESSORS</td>
<td>Warehouse Advantage Processors</td>
<td>Manages a set of background processes for Warehouse Advantage. There is no user interface to this solution environment.</td>
</tr>
<tr>
<td>WA</td>
<td>Warehouse Advantage</td>
<td>Manages inventory within the four walls of the warehouse</td>
</tr>
</tbody>
</table>
Scenario: You want to promote a new application into the AFA solution environment. Due to the 24/7 nature of your business, you are unable to find a time when all of the users are off the system. You want to disrupt as few individuals as possible during the promotion.

Procedures: The procedures below demonstrate how to stop and start a single solution environment. However, they do not demonstrate the promotion process.

☐ Open the AWESM tool, if necessary.

☐ Start the engine, if necessary.

☐ Open the Virtual Terminal, if necessary.

☐ Click the AFA solution environment in the AWESM tool in order to highlight it.

☐ Click the Stop button in the lower portion of the window.

The system displays a dialog asking if you want to stop the AFA solution environment.

☐ Click the OK button.

The system changes the AFA solution environment icon from green to yellow to red, and it changes the state of the AFA solution environment to "Stopped". However all of the other solution environments in the AWESM tool remain unchanged.

Additionally, the main screen of the Virtual Terminal still shows the USER ID prompt.
At this point, you could promote the AFA application into production without negatively impacting the RF users associated with WA solution environment. The next couple steps demonstrate that the WA solution environment is still working, even though the AFA solution environment is shut down.

- Log into the Virtual Terminal with a user id, a password, and a forklift.
- Press the F1 key on the Virtual Terminal until you return to the USER ID prompt.

The AFA scenario does not require you to shut down the WA solution environment. However, the next couple steps prove that you can shut down multiple solution environments at the same time.

- Click the WA solution environment in the AWESM tool in order to highlight it.
- Click the Stop button in the lower portion of the window.

The system displays a window asking if you want to stop the WA solution environment.

- Click the OK button.

The system changes the WA solution environment icon from green to yellow to red, and it changes the state of the WA solution environment to “Stopped”. However all of the other solution environments in the AWESM tool remain unchanged.
Additionally, the main screen of the Virtual Terminal indicates that the engine is down. Technically, it is more accurate to say that the WA solution environment is down. In either case, the end users can no longer use the RF terminals in conjunction with the WA solution environment.

After you have promoted the new AFA application into production, you need to restart the AFA solution environment so the individuals performing the wave release can continue their work.

☐ Click the AFA solution environment in the AWESM tool in order to highlight it.

☐ Click the Start button in the lower portion of the window.

The system changes the AFA solution environment icon from red to green, and it changes the state of the AFA solution environment to “Running”. However, all of the other solution environments in the AWESM tool remain unchanged.

At this point the individuals performing the wave release can continue their work. Although the AFA scenario did not require it, you also shut down the WA solution environment above. You need to restart the WA solution environment as well.

☐ Click the WA solution environment in the AWESM tool in order to highlight it.

☐ Click the Start button in the lower portion of the window.
The system changes the WA solution environment icon from red to yellow to green, and it changes the state of the WA solution environment to "Running". However, all of the other solution environments in the AWESM tool remain unchanged.

<table>
<thead>
<tr>
<th>Solution Environment(s)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADVANTAGEPLATFORM</td>
<td>Running</td>
</tr>
<tr>
<td>ADVUNIFLATILE</td>
<td>Running</td>
</tr>
<tr>
<td>AFA</td>
<td>Running</td>
</tr>
<tr>
<td>WA</td>
<td>Running</td>
</tr>
<tr>
<td>WAPROCESSORS</td>
<td>Running</td>
</tr>
</tbody>
</table>

Additionally, the main screen of the Virtual Terminal is prompting for a USER ID. The users with terminals connected to the Warehouse Advantage solution environment can continue their work.
Activity 7: Stop the Engine with Active Users

Background: As mentioned before, shutting down the engine of a production system is not something that should be treated lightly. In an ideal situation, all of the users would log off the system before you shut down the engine. However, sometimes users forget to log off their terminals. This places you in the precarious position of determining if you should shut down the engine while users are actively using the system.

Scenario: You want to shut down the engine. However, there are active users on the system. You want to gather a little more information about these users before making a decision about an engine shutdown.

Procedures: The procedures below demonstrate how to review information related to active users. They also demonstrate how to shut down the engine with active users.

☐ Start all solution environments, if necessary.

☐ Open the Virtual Terminal, if necessary.

☐ Log into the Virtual Terminal with a user id, a password, and a forklift.

The AWESM tool indicates that there is one active WA session, namely the virtual terminal from above.

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>Terminals</th>
<th>Sessions</th>
<th>Version</th>
<th>Activation Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADVANTAGE PLATFORM</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2016-07-20 16:24:19</td>
</tr>
<tr>
<td>ADVUNIFLATE</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2016-08-08 10:55:51</td>
</tr>
<tr>
<td>AFA</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2016-08-08 15:19:48</td>
</tr>
<tr>
<td>WA</td>
<td>Running</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>2016-08-17 17:05:40</td>
</tr>
<tr>
<td>WAPROCESSORS</td>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2016-08-08 15:06:17</td>
</tr>
</tbody>
</table>

☐ Click the WA solution environment in order to highlight it.

☐ Click the Sessions button.

The system opens another window which displays all of the sessions for the WA solution environment. This window shows terminals (RADIO_01), and it shows other “system-type” sessions. These “system-type” sessions are essentially sessions running in the background.
When you want to shut down the engine, you are most likely primarily concerned with the terminals. The next instruction demonstrates how to filter this list so that it only displays information related to terminals.

☐ Choose the **Show Active Devices** in the **View** drop-down list.

The system filters the above list, and only displays active terminals.

The system indicates that there is one active terminal (RADIO_01), and that AMY is the one who is logged into the terminal. It also indicates that AMY is currently in a process object called WA.Menu. While the Process Object value holds little meaning for an administrator, it holds great significance to a developer who is familiar with the Advantage Architect tool. In this case, it simply means that AMY is on screen which displays a list of business process menu options. If AMY had been in the middle of a picking process, the consequences of shutting down the engine might be severe. But since she is sitting on the main menu, you can shut down the engine without impacting her work.

☐ Click the **X** in the **upper right corner** of the Sessions window.

☐ Click the **WA solution environment** in order to highlight it.

☐ Click the **Stop** button in the lower portion of the window.

The system displays a window asking if you want to stop the WA solution environment.
Click the **OK** button.

Because there are active sessions in the WA solution environment, the system does not automatically stop it. Instead it places the WA solution environment into a Stop Pending state.

In a Stop Pending state, the current users can continue to run business processes from the terminals. But once all active sessions to the solution environment are gone, the system will automatically shut down the engine. The solution environment will remain in this pending state until one of the following occurs:

- An administrator forces the stop
- An administrator cancels the stop
- All users exit their WA sessions.

In this example, you will force the engine to stop even though there are active sessions.

Click the **WA solution environment** in order to highlight it.

Click the **Force Stop** button.

The system gives one final notification before you stop a solution environment with active users.
☐ Click the **OK** button.

At this point, the system shuts down the WA solution environment, even though there are active users on the system.

<table>
<thead>
<tr>
<th>Solution Environment(s)</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADVANTAGE PLATFORM</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>ADVLINK_FLATFILE</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>APA</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>WA</td>
<td>Stopped</td>
</tr>
<tr>
<td></td>
<td>WAPROCESSORS</td>
<td>Running</td>
</tr>
</tbody>
</table>
Admin Lab Exercise 2:  
Common Tasks in the Training Activities

Introduction

This exercise consists of the following activities:

1. Determine the Machine Name
2. Run the Location Status Business Process
3. Determine the User ID and Password for an Application

Background Information

As you work your way through the remaining exercises, you will find some commonalities among them. Many of the exercises ask you to type the machine name into an edit box; some require familiarity with a specific business process called Location Status; and most require you to log into an application with a user name and a password.

Rather than repeating the procedures for each of these common tasks multiple times, the procedures are only detailed once in the entire training manual. They are detailed in this specific exercise. All of the exercises that follow this one expect that you know how to perform the common tasks detailed here.
Activity 1: Determine the Machine Name

Background: In a production environment, most HighJump customers have three different servers: an application server, a web server, and a database server. Some customers have additional servers, and others have fewer. The chart below briefly describes the purpose of each.

<table>
<thead>
<tr>
<th>Server</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>Runs the Advantage Workflow Engine</td>
</tr>
<tr>
<td>Web Server</td>
<td>Serves up web pages to web browsers</td>
</tr>
<tr>
<td>Database Server</td>
<td>Houses SQL Server or Oracle</td>
</tr>
</tbody>
</table>

In this training environment, there is only one server. The application server, the web server, and the database server are all rolled into one machine. This is an important distinction to keep in mind as you compare this environment with your production environment.

Many applications in the Advantage system require you to enter a machine name at some point in the process. Virtual Terminal is one example. In some cases, you can substitute the keyword LOCALHOST to indicate the machine on which the application resides. In other cases LOCALHOST does not work properly, and the true machine name must be used.

Situation: You are in the process of defining a Web Terminal in the Advantage system. A Web Terminal is similar to a Virtual Terminal, except that it runs within Internet Explorer. As part of the definition process, the system asks you to enter the machine name of the application server.

Procedures: The procedures below demonstrate how to determine the machine name of the computer. However, they do not demonstrate how to define a Web Terminal. That is covered in a later exercise.

☐ Click the Start | Server Manager menu option.

The system opens the Server Manager application.

☐ Click the Local Server option in the panel on the left.
The system displays several settings for the given server. In the upper left corner of the properties, the system displays the computer name of the given server.

☐ Write the **computer name** here __________________________.

From this point forward, whenever a procedure in this manual asks you to enter the computer name (also known as the machine name) of the application server, the web server, or the database server, you should enter the computer name you captured above.

☐ Close the **Server Manager** window.

The system closes the windows you opened during this activity.
Activity 2: Run the Location Status Business Process

**Background:** In the previous activity, you learned how to log on to the Virtual Terminal and navigate to the main menu. While the logon process provides a little bit of background activity, several components of this manual benefit from running a business process that is more substantial in nature. The business process called Location Status will serve this purpose.

In the Location Status business process you can query and modify the status of a location in the warehouse. For example, if a water pipe breaks and floods a couple locations, you may want to temporarily deactivate the locations in the HighJump system until the water is removed. You can use the Location Status business process to deactivate the water-logged locations.

Each location in the HighJump system has one of four different statuses. They are listed in the chart below. If you modify the status of a location with the Location Status business process, you must use one of these values.

<table>
<thead>
<tr>
<th>Status Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Empty</td>
</tr>
<tr>
<td>P</td>
<td>Partially Full</td>
</tr>
<tr>
<td>F</td>
<td>Full</td>
</tr>
<tr>
<td>I</td>
<td>Inactive</td>
</tr>
</tbody>
</table>

**Scenario:** A forklift driver accidentally ran into two locations and damaged them. You want to temporarily deactivate the damaged locations so that the HighJump system does not direct anybody to them. You decide to use the Location Status business process to deactivate them.

**Procedures:** The procedures below demonstrate how to use the Location Status business process to change the status of a location in the warehouse

- Start the **engine**, if necessary.
- Start the **Virtual Terminal**, if necessary.
- Log into the Virtual Terminal with a user, a password, and a forklift.

The system displays the main menu for the user.
The menu is just one place among many where the HighJump system displays a list of options on a terminal. During this class, you may encounter other places in the application that utilize a list. Anytime the system displays a list on a terminal there are two function keys you can use to navigate through the list. Due to space limitations, these function keys are not described on the terminal screen. The chart below describes the function keys and their purposes.

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4</td>
<td>Page Up (previous page of data)</td>
</tr>
<tr>
<td>F8</td>
<td>Page Down (next page of data)</td>
</tr>
</tbody>
</table>

- Press the **F8 key** in order to page down to the next set of options. (This step is optional.)
- Enter **7 (Inv Control)** at the **Option** prompt.
- Enter **3 (Inventory Status)** at the **Option** prompt.
- Enter **4 (Location Status)** at the **Option** prompt.

The system advances to the first prompt of the Location Status business process.
There are several locations that will work at this prompt for the purpose of training. The screenshot below depicts several options. You can find other valid locations by looking at the HighJump One Platform User Interface (UI) pages.

<table>
<thead>
<tr>
<th>#</th>
<th>Warehouse ID</th>
<th>Location ID</th>
<th>Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01</td>
<td>M101</td>
<td>Multiple Item Storage</td>
<td>Empty</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>M102</td>
<td>Multiple Item Storage</td>
<td>Empty</td>
</tr>
<tr>
<td>3</td>
<td>01</td>
<td>M103</td>
<td>Multiple Item Storage</td>
<td>Empty</td>
</tr>
</tbody>
</table>

- Enter **M120** at the **Location** prompt.

The system displays the current status of the given location, and then it prompts for a new status.
Enter I at the Location Status prompt.

The system accepts the new status and then returns to the top of the business process. It does not give an explicit message indicating that it changed the status. It simply goes back to the top of the process. (If you want to validate that the status was changed, then enter M120 a second time at the Location prompt, and the system will display the new status.)

Enter M121 at the Location prompt.

Enter I at the Location Status prompt.

Once again, the system accepts the new status and then returns to the top of the business process. It does not give an explicit message indicating that it changed the status. It simply goes back to the top of the process.

As a general rule, anytime you want to navigate backwards in the prompting sequence (or if you want to navigate backwards to the main menu) you can press the F1 key. Due to space limitations, this function key is not described on the terminal screen. There are plenty of times when F1 is not a valid option, and the system will display an error if you press the F1 key in those circumstances. There are many times during this class when you will need to exit out of a business process or navigate backwards to the User ID prompt. Use the F1 key to perform those operations.

Press the F1 key multiple times until you return to the main menu.

At this point, you could choose another business process if desired.

Press the F1 key multiple times until you return to the User ID prompt.

From this point forward, all of the training exercises assume that you have a working knowledge of the Location Status business process. Use this exercise as a reference if there are some details that you cannot remember when you encounter those exercises.
Activity 3: Determine the User ID and Password for an Application

**Background:** Several of the applications you will encounter during this training class require you to log in with a user id and a password. This includes HighJump applications as well as other applications. All of the user ids and passwords that you will need for this class are listed in a centralized location.

The user ids and passwords are specific to the training class. It is likely that they will be different in the environments at your facility. Additionally, the centralized list is specific to the training class. The list is not part of the standard installation.

**Scenario:** You want to review a list of all employees logged into RF readers during the training class. In order to get this information, you open the HighJump One Platform User Interface (UI). The first page it displays is a page that requests a user id and a password. You want to know the user id and password for the HighJump One Platform UI.

**Procedures:** The procedures below demonstrate how to find a user id and password for an application during the training class.

- Double click the Training folder on the desktop.
- Double click the Misc folder.
- Double click the login.txt file.

The system opens a text file that contains a list of all the user ids and passwords that you will use throughout the training class. The user ids and passwords are broken down by application. The user id is on the first line of each section. The password is on the second line of each section. In this case, the user id for SQL Server is “sa”, and the password is “sapass#1”

- Scroll down to a section titled HighJump One Platform UI #1.

The document indicates that the user id is “Administrator” and the password is “HJSPASS” for the HighJump One Platform UI.
Choose the **File | Exit** menu.

From this point forward, the training exercises do not explicitly state the user ids and passwords for a given application. Instead they use language similar to the following: “Use the login.txt file in the folder on the desktop to populate the user id and password edit boxes.” The login.txt file is the one that is described in this activity.
Admin Lab Exercise 3: 
User Profiles and Forklifts

Introduction

This exercise consists of the following activities:

1. Create a Material Handler User
2. Create a HighJump One Platform User
3. Add a New Forklift
4. Change the Menu Level of a Material Handler User
5. Change the Menu Level of a HighJump One Platform User

Background Information

Prior to going live with Warehouse Advantage, several components of the warehouse must be defined in the system. This includes, but is not limited to, locations, zones, items, employees, vendors, and customers. Some of these elements are defined in a host system and then passed to the HighJump system via some interface. Other elements are defined only in the HighJump system. While employees (material handlers) and forklifts are sometimes defined in a host system and passed to WA, they are more commonly defined in WA and nowhere else. The activities below demonstrate how to add an employee (material handler) and a forklift to the HighJump system through the HighJump One Platform User Interface (UI).

In addition to accessing the RF terminals, several users will need to access the HighJump One Platform UI pages in order to view and update data. The activities below demonstrate how to add such user to the system.

Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Add Users, Forks, and Handlers menu as well as the Create HJ1 User Profile menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Create a Material Handler User

**Situation:** The warehouse manager calls you on the phone. He explains that the company has been doing very well financially. The volume of the inbound and outbound orders has dramatically increased over the last several months. In order to manage this increase in activity, he has decided to hire a couple of individuals to work on the receiving dock. Max Brimmer is one of these new employees, and he starts next Monday. The warehouse manager asks you to create a profile for Max so he can execute receiving workflows on an RF terminal.

☐ Click the **Menu | Supply Chain Advantage | Warehouse Advantage | Employees | Employees | menu.**

The system displays the Search Employees page.

☐ Click the **Query button in the Action Bar.**

The system displays a list of all current employees (material handlers).

☐ Click the **Add Employee button in the Action Bar.**

The system displays the Edit/Add Employee page.
This page is where you will enter Max’s settings.

- Choose 01 – Warehouse 01 in the Warehouse ID drop down box.
- Type MAX in the Employee ID edit box.
- Type MAX BRIMMER in the Employee Name edit box.
- Type MAX in the Password edit box.

The Menu Level option dictates which menus the user will see when he logs into an RF terminal. RECEIVER is the menu level in the sample data delivered with the base application that gives the end user the ability to perform receiving business processes.

- Choose RECEIVER in the Menu Level drop down box.
- Click the Insert button in the Action Bar.

The system returns to the Employee page and displays Max Brimmer’s profile.

The final step in this process is to validate that the changes work as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.
The test cases below do not detail every keystroke. You need to use the knowledge you gained in previous activities to test these scenarios.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Open a Virtual Terminal and type <strong>MAX</strong> at the USER ID prompt.</td>
<td>System advances to the PASSWORD prompt.</td>
</tr>
<tr>
<td></td>
<td>Type <strong>MAX</strong> at the PASSWORD prompt.</td>
<td>System advances to the EQUIPMENT / ZONE prompt.</td>
</tr>
</tbody>
</table>
Activity 2: Create a HighJump One Platform User

**Background:** Anyone who wants access to the HighJump One Platform UI must have a HighJump One Platform profile defined in the HighJump Workspace tool. The profile includes the user name, password, culture, and a set of parameters that dictate which pages the user can access. This activity focuses on creating the overall profile. The permission-related parameters are a mandatory part of the profile, and they will be discussed at length in a later activity.

**Scenario:** Jason Meyers is a new hire, and he is going to start his new role in the warehouse next week. It is your responsibility to grant Jason access to the HighJump One Platform UI. Because he will operate in an administrative capacity, he needs full access to all pages.

*Perform the following steps to create a new HighJump One Platform User.*

- Open another instance (or another tab) of **Internet Explorer**.
- Enter the following URL in the address bar.
  
  `http://[web_server]:30000/workspace/default.html`

The system displays the login screen.

- Choose **Workspace Authentication** in the **Authentication** drop-down list, if necessary.
- Use the HighJump One Workspace section of the `login.txt` file in the folder on the **desktop** to populate the **User Name** edit box and the **Password** edit box.
- Click the **Login** button.

The system displays the main page of the HighJump One Workspace.
☐ Click the **Administration | Users** menu.

The system adds a Users tab to the interface.

It also displays a list of all HighJump One Platform users in the bottom portion of the screen.

☐ Click the **Add** button in the **Action Bar**.

The system opens the User window which allows you to configure all of the user-related settings.
The Login setting is the User Name that the user enters when he logs into the HighJump One Platform UI.

☐ Enter **Jason** in the **Login** edit box.

The system displays the First Name and the Last Name settings in the upper right corner of the HighJump One Platform UI after the user logs into it.

☐ Enter **Jason** in the **First Name** edit box.

☐ Enter **Meyers** in the **Last Name** edit box.

The Email setting is a required value, and the HighJump One Platform uses it to communicate with the user for various reasons. If the user has no address, then you can populate it with a dummy email.

☐ Enter **jason.meyers@highjump.com** in the **Email** edit box.
The New Password setting and the Confirm Password setting both reflect the password the user enters when he logs into the HighJump One Platform UI.

- Enter **HJSPASS** in the **New Password** edit box.
- Enter **HJSPASS** in the **Confirm Password** edit box.

The HighJump One Platform has the ability to display the web page structural components (page headers, column headers, button titles, etc.) in multiple languages. Which language it uses is dependent upon the user who logged into the HighJump One Platform User Interface. The HighJump One Workspace allows you to associate a user with a language or culture. In the training environment the only language that is supported is English. Other languages are available, but they must be installed with a separate language pack. In this activity you will explicitly associate Jason with the English culture—even though this is unnecessary as the rules default to English in the training environment.

- Click the **User Interface** tab.

The system displays several settings related to the UI.

- Choose **English (United States)** in the **Culture** drop-down list.
A role reflects which responsibilities this user will hold in the warehouse. A role in the HighJump One Workspace corresponds directly with a User Group in Page Editor. Roles and User Groups are related to permissions. Together, they answer the question, “Who has access to which pages?” Roles and User Groups will be discussed in more detail later in this manual.

☐ Click the Roles tab.

The system displays a list of all roles.

In this example, Jason will use the HighJump One Platform UI to perform administrative capabilities. You will assign him to the Administrators.mPage role. (The Administrators role is not used in conjunction with the HighJump One Platform User Interface.)

☐ Click and hold the Administrators.mPage role in the All Roles list, in order to select it.

☐ Drag and drop the selection to the User Roles list on the right.

The system moves the Administrators.mPage role from the list on the left to the list on the right.
The HighJump One Workspace allows you to grant or revoke access to an application at the user level or at the role level. If you do not specify the access at the user level, then the user permissions will be inherited from the roles to which the user belongs. In this VLab, all privileges to the applications have been defined at the role level, and the users inherit the privileges from the roles.

In the steps below you will not change the permissions, but you will see how to determine the permissions for a user without directly reviewing the role definition.

☐ Click the Authorization tab.

The system displays the three applications.

By default all of the authorization settings are set to Not Declared. Not Declared indicates that the authorization to the given application is inherited from the role membership.

☐ Hover the cursor over the Advantage Commander 12.7 application.

The system pops up a window showing all of the roles to which Jason belongs. For each role, the system indicates if the role allows or denies privileges to the selected application. In this example, Jason is a member of the Administrator.mPage role, and that role grants access to the Advantage Commander 12.7 application.

☐ Hover the cursor over the Supply Chain Advantage 12.7 application.

Once again, the system pops up a window showing all of the roles to which Jason belongs. Because of his membership in the Administrators.mPage roles, Jason will also have access to the HighJump One Platform 12.7 application.
If you repeat the steps for the HighJump One Platform 1.0 application will see similar results.

A Menu Level controls which menu options the user sees in the menu panel after he logs into the HighJump One Platform UI. The term “Menu Level” is common to both the HighJump One Workspace tool and the Page Editor tool. In this example, Jason will use the HighJump One Platform UI to perform administrative capabilities. You will assign him to the Administrator menu level for all applications.

- Click the **Identity Claims** tab.

The system displays a menu level setting and a WW_USERID settings for both Advantage Commander and Supply Chain Advantage.

- Choose **Administrator** in the **Menu Level** drop-down list for the Advantage Commander application.
- Choose **Administrator** in the **Menu Level** drop-down list for the Supply Chain Advantage application.
- Click the **Save** button in the lower right corner of the window.

The system returns to the master list HighJump One Platform users. However, Jason’s record is not visible on the first page data. You need to advance to the second page of data to see his profile.
☐ Click the **Next Page** button in the lower left corner of the window.

The system now displays Jason in the list. You may need to scroll down to see his record.

At this point, the definition of Jason’s profile is complete. The steps below demonstrate how to test your changes in the HighJump One Platform UI.

☐ In the HighJump One Platform UI click the **Administrator | Logout** menu option near the upper right corner.

The system displays a logout confirmation screen.

☐ Click the **Logout** button.

The system displays the login screen.
Enter **Jason** in the **User Name** edit box.

Enter **HJSPASS** in the **Password** edit box.

Click the **Login** button.

The system opens the HighJump One home page and displays Jason’s name in the upper right corner.

Click the **Application Menu** button in the **Menu Bar**.

The system displays the Advantage Commander and the Supply Chain Advantage menu options. From here, you can navigate through either one of these menu options.
Activity 3: Add a New Forklift

Situation: The warehouse manager calls you on the phone. He explains that they have added some new storage racking in the warehouse that extends 6 tiers upwards toward the ceiling. In order to access the material in the upper tiers, the company has purchased a new “cherry picker” forklift. The warehouse manager asks you to add a forklift to the WA system for this new “cherry picker” so that the material handlers can use it in the business processes.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Supply Chain Advantage | Warehouse Advantage | Warehouse Setup | Locations menu.

The system displays the Search Location page.

☐ Choose Forklift in the Location Type drop-down list.

The above step is not required in order to add a new forklift. However, if you do not provide any filter (search) criteria, the system will display a list of all locations across all warehouses. Due to the large amount of data, the system may take a long time to display the list. By providing a location type in the drop-down list, the amount of returned data is smaller, and the system will display the data set in a shorter time period.

☐ Click the Query button in the Action Bar.

The system displays a list of all defined forklifts across all warehouses.
Click the Add New Location button in the Action Bar.

The system displays the Edit/Add Location page.

Choose Warehouse 01 in the Warehouse ID drop-down list.

Type CP001 in the Location ID edit box.

Choose Forklift in the Type drop-down list.

Type Cherry Picker 1 in the Description edit box.

Click the Insert button in the Action Bar.

The system now lists CP001 as a forklift location.
The final step in this process is to validate that the changes work as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

The test cases below do not detail every keystroke. You need to use the knowledge you gained in previous activities to test these scenarios.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Open a Virtual Terminal and login with a valid user and password</td>
<td>System advances to the <strong>EQUIPMENT / ZONE</strong> prompt.</td>
</tr>
<tr>
<td></td>
<td>Type <strong>CP001</strong> at the <strong>EQUIPMENT / ZONE</strong> prompt.</td>
<td>System advances to the main menu.</td>
</tr>
</tbody>
</table>
Activity 4: Change the Menu Level of a Material Handler User

**Situation:** The warehouse manager calls you on the phone. He explains that Paul normally performs packing activities. However, due to several illnesses in the receiving department, the manager wants Paul to work in the receiving department for one day only. The manager fully anticipates that the receiving department will be fully staffed tomorrow, and that Paul could resume his normal packing activities at that time. Because Paul’s current profile does not allow him to run the receiving business processes on the RF terminal, the manager asks you to grant Paul access to these processes.

The first couple steps demonstrate the current situation. They demonstrate that Paul’s current profile does not allow him to run the receiving business processes.

- Open a Virtual Terminal.
- Use the Virtual Terminal #1 section of the login.txt file to log into the Virtual Terminal.

The system displays the main menu for Paul. The only two types of activities he can execute are packing processes and report generation.

In a previous activity, you used the Warehouse Advantage web application to create a new RF terminal user profile. In this activity, you will use the same tool in order to change Paul’s RF terminal menu.

- Open the HighJump One Platform UI, if necessary.
- Click the Menu | Supply Chain Advantage | Warehouse Advantage | Employees | Employees menu.

The system displays the Search Employees page.
Choose 01 - Warehouse 01 in the Warehouse ID drop-down list.

Choose 01-PAUL PACKER in the Employee ID drop-down list.

Click the Query button in the Action Bar.

The system returns the specified user profile.

Click the number to the left of the Employee Name attribute.

The system opens the Add/Edit User – Warehouse Advantage page which displays all of the settings related to the RF terminal side of Paul’s profile.
Choose **RECEIVER** in the **Menu Level** drop-down list.

The settings should look like the following screenshot.

- **Menu Level**: RECEIVER
- **Language**: English (United States)

Click the **Update** button in the **Action Bar**.

The system displays Paul's updated profile.

The changes related to the RF terminal side of his profile take effect immediately. There is no need to publish the profiles. All of the configuration is complete. The final step in this process is to validate that the changes work as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely...
that you either missed a step or that you performed a step incorrectly. The test cases below do not detail every keystroke. You need to use the knowledge you gained in previous activities to test these scenarios.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Log out of the Virtual Terminal, and then log in a second time as <strong>PAUL</strong>.</td>
<td>The system displays the receiving menu on the terminal.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image.png" alt="Screenshot" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PAUL PACER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Receipts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Putaway</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 Movement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPTION</td>
</tr>
</tbody>
</table>
Activity 5: Change the Menu Level of a HighJump One Platform User

Situation: The warehouse manager calls you on the phone. He explains that Ron, who builds and releases waves for picking, has been given a promotion. Ron currently has access to several of the Advanced Fulfillment pages. However, in his new position, he will need the ability to perform some warehouse setup tasks. This includes the ability to add new forklifts and modify the statuses of stock locations. The warehouse manager asks you to grant Ron permissions to the Warehouse Advantage application in the HighJump One Platform UI.

The first couple steps demonstrate the current situation. They demonstrate that Ron’s HighJump One Platform profile allows him to access the Advanced Fulfillment pages, but not the Warehouse Advantage pages.

☐ In the HighJump One Platform UI, choose the Administrator | Logout menu in the upper right corner.

The system displays a logout confirmation screen.

☐ Click the Logout button.

The system returns to the login screen.
Use the **HighJump One Platform UI #3** section of the `login.txt` file to log into the HighJump One Platform UI.

Click the **Menu | Supply Chain Advantage** menu.

The system displays Ron’s menu. The only application he can access is Advanced Fulfillment Application.

The next several steps demonstrate how to change Ron’s menu level and give him access to additional HighJump One Platform UI applications.

Log into the **HighJump One Workspace**, if necessary.

Click the **Administration | Users** menu.

The system displays a list of HighJump One Platform users.
Click the **Next Page** button two times near the lower left corner of the window.

The system advances to the third page of HighJump One Platform users.

Double-click the row where the **Login** value is **RON**.

The system opens a window which allows you to modify Ron’s user settings.

Click the **Identity Claims** tab.

The system displays a menu level setting and a WW_USERID settings for both Supply Chain Advantage and Advantage Commander.
Expand the **Supply Chain Advantage** options.

Ron’s menu level in the Supply Chain Advantage suite is currently set to User. This grants him limited access to the Advanced Fulfillment application. The Administrator menu level has far greater permissions than the User menu level.

Choose **Administrator** in the **Menu Level** drop-down list for the **Supply Chain Advantage** application.

You switched Ron’s menu to that of an Administrator. However, his current Role setting which drives the permissions to the pages will prevent him from visiting many additional menu options included on the Administrator menu. In order to grant him access to the additional pages, you also need to modify his Role setting.

Click the **Roles** tab.

The system indicates that Ron is a member of the role called Users.

Click and hold the **Managers** role in the **All Roles** list, in order to select it.

Drag and drop the selection to the **User Roles** list on the right.
Ron is now a member of both the Users role and the Managers role.

- Click the **Save** button in the lower right corner of the window.

The system returns to the master list of HighJump One Platform users.

At this point, you have changed Ron’s menu level and his role in the HighJump One Workspace. The next several steps below demonstrate how to test your changes in the HighJump One Platform UI. Because this modification has an effect on the menu panel, you must logout and login in order to see the changes.

- In the HighJump One Platform UI, **logout** of the application.

- Use the **HighJump One Platform UI #3** section of the **login.txt** file to log into the HighJump One Platform UI.

- Click the **Menu | Supply Chain Advantage** menu.

The system displays Ron’s menu. Ron can now access the Advanced Fulfillment and Warehouse Advantage applications.
In the above procedures, you modified the Menu Level setting and the Role in the HighJump One Workspace to change the menu options available in the HighJump One Platform UI. There are other settings that can also control which menu options the end user sees in the HighJump One Platform UI, but those are outside the scope of this class.
Admin Lab Exercise 4:
Terminal Menus

Introduction

This exercise consists of the following activities:

1. Explore the RF Menu Definitions in HighJump One Platform UI
2. Modify the RF Menus

Background Information

After a material handler logs into an RF terminal with a user, a password, and an equipment id, the system presents a menu on the terminal. This menu is highly configurable. Some customers configure the menu structure by writing SQL statements against the menu master table in the AAD database. Others configure the menu by interacting with a series of web pages. In this exercise, you will make several modifications to a menu by using the web-based interface.

Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Modify RF Menu menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Explore the RF Menu Definitions in HighJump One Platform UI

Scenario: Your company has decided to make some significant changes to the menus that appear on the RF terminal. You want to explore the existing menus before you make any changes. In this activity you will not make any changes to the system. Rather, you will explore the relationship between the menu on the RF terminal and the data in the HighJump One Platform User Interface (UI) pages.

First, you will make note of the top-level menu options that AMY sees when she logs into a terminal.

☐ Start the engine, if necessary.

☐ Log in to the Virtual Terminal as AMY / AMY / FAAMY.

☐ Press the F8 key to scroll down a page.

The following screenshots show all top-level menu options available to AMY.

Each RF user is associated with a menu level that dictates which menu options she sees and the order in which they appear on the screen. Next you will determine the menu level to which AMY is assigned.

☐ In the HighJump One Platform UI, logout of the application.

☐ Use the HighJump One Platform UI #1 section of the login.txt file to log into the HighJump One Platform UI.

☐ Click the Menu | Supply Chain Advantage | Warehouse Advantage | Employees | Employees menu.

The system opens the Search Employees page.
Click the **Query** button in the **Action Bar**.

The system displays a list of all employees for the HighJump system.

<table>
<thead>
<tr>
<th>#</th>
<th>Employee Name</th>
<th>Employee ID</th>
<th>Warehouse ID</th>
<th>Menu Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AMY ADMINISTRATOR</td>
<td>AMY</td>
<td>01</td>
<td>BASE</td>
</tr>
<tr>
<td>2</td>
<td>CURT CUSTOMER</td>
<td>CURT</td>
<td>01</td>
<td>BASE</td>
</tr>
<tr>
<td>3</td>
<td>ED EXECUTIVE</td>
<td>ED</td>
<td>01</td>
<td>BASE</td>
</tr>
</tbody>
</table>

In the middle of this page, the system displays the Menu Level for each employee (RF user). Note that AMY’s Menu Level is set to BASE. “BASE” is not a HighJump keyword. It is a menu level included in the sample data set which allows the employee to launch any business process included in the base application.

Next, you will use this information to view the BASE menu level as it is defined in the web pages.
☐ Click the **Menu | Supply Chain Advantage | Warehouse Advantage | System Administration | RF Menu Management** menu.

The system displays the Search Menu page.

In this example, AMY is an English speaker, so you can leave the Locale ID as English. Based upon the previously-visited screens, you know that AMY’s menu level is BASE, so you leave the Menu Level as-is. The Name setting is an identifier for the menu or submenu. “MAIN” is a HighJump keyword. It refers to the top-level menu on the terminals. All of the default settings are in line with the data we want to view.

☐ Click the **Query** button in the **Action Bar**.

The system displays the same top-level menu that AMY sees when she logs into a terminal.

Each of the top-level menu options on AMY’s terminal will display a submenu.

☐ On the **Virtual Terminal**, enter **3 (Replenishment)** at the **Option** prompt.

☐ In the **HighJump One Platform UI**, click the **Sub Menu** link associated with the **Replenishment** row.
Once again, the web page shows the same menu that AMY sees on the terminal.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Text</th>
<th>Sub Menu</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Request TopOff</td>
<td></td>
<td>Request TopOff Replenishment</td>
</tr>
<tr>
<td>2</td>
<td>Move</td>
<td></td>
<td>Move</td>
</tr>
<tr>
<td>3</td>
<td>Request Item</td>
<td></td>
<td>Request an Item</td>
</tr>
<tr>
<td>4</td>
<td>Kitting/Unkitting</td>
<td>KITTING</td>
<td></td>
</tr>
</tbody>
</table>

In the HighJump One Platform UI, the Text column is the value that is displayed on the terminal. The Sub Menu column (if populated) will link to the contents of the sub menu. The Process column (if populated) points to the name of the Advantage Architect process object which the system will launch when the user chooses that menu option on the RF terminal. Each menu option on the RF terminal will either drill into a sub menu or it will launch a business process. Consequently, in the HighJump One Platform UI, you will never see a row with both the Sub Menu and the Process columns populated. Either the Sub Menu column or the Process column will be populated – but not both. In the example menu from above, options 1, 2, and 3 will launch a specific business process, and option 4 will drill into a submenu. With this in mind, you will now begin to modify the structure of this menu in the next activity.
Activity 2: Modify the RF Menus

Situation: The warehouse manager asks you to make several changes to the replenishment menu you reviewed in the previous activity. He asks you to rename some options; reorder the options; remove some options; and add a Cycle Count submenu. The diagram below shows the modified replenishment menu structure and the corresponding cycle count submenu which he wants you to create.

Task: Rename the “Kitting/Unkitting” menu to “Kits”

1. In the HighJump One Platform UI, click the number to the left of the row where the Text column is Kitting / Unkitting.

The system opens the Edit / Add Menu window.

2. Enter Kits in the Text edit box.

3. Click the Update button in the Action Bar.

The system returns the Replenishment menu page and shows the updated text.
You have made the change in the HighJump One Platform UI. Next you will test the changes on the terminal. In order to see the change on the terminal, you need to navigate away from the current screen and then return to it.

- On the Virtual Terminal, press the F1 key to navigate back to the main menu.

- Enter 3 (Replenishment) at the Option prompt.

The system displays the Replenishment menu on the terminal with the updated menu options.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Text</th>
<th>Sub Menu</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Request TopOff</td>
<td>Request TopOff Replenishment</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Move</td>
<td>Move</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Request Item</td>
<td>Request an Item</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Kits</td>
<td>KITTING</td>
<td></td>
</tr>
</tbody>
</table>

**Task:** Move the Request TopOff option down to position number 2 and the Kits option up to position 3

- In the HighJump One Platform UI, click the Move Down link for the row where the Text column is Request Topoff.

- Click the Move Up link for the row where Text column is Kits.
The system returns the Replenishment menu page and shows the modified structure.

You have made the change in the HighJump One Platform UI. Next you will test the changes on the terminal. In order to see the change on the terminal you need to navigate away from the current screen and then return to it.

☐ On the Virtual Terminal, press the **F1 key** to navigate back to the main menu.

☐ Enter **3 (Replenishment)** at the **Option** prompt.

The system displays the Replenishment menu on the terminal with the updated menu options.

```
Replenishment
1 Move
2 Request TopOff
3 Kits
4 Request Item
```

**Task:** Remove the Request Item option from the menu.

☐ In the HighJump One Platform UI, click the **Delete** link for the row where the **Text** column is **Request Item**.

The system asks you to confirm the delete action.
Choose Yes in the Confirm Delete drop-down list.

Click the OK button in the Action Bar.

The system returns the Replenishment menu page and shows the modified structure.

You have made the change in the HighJump One Platform UI. Next you will test the changes on the terminal. In order to see the change on the terminal you need to navigate away from the current screen and then return to it.

On the Virtual Terminal, press the F1 key to navigate back to the main menu.

Enter 3 (Replenishment) at the Option prompt.

The system displays the Replenishment menu on the terminal with the updated menu options.
**Task:** Add a Cycle Count submenu with two options underneath it.

In this task you will add a submenu to position 4 on the Replenishment menu which reads “Cycle Counts”. When you select this menu option from the RF terminal, the system will display a menu containing an option which launches the Cycle Count Check business process and another option which launches the Cycle Count business process. The diagram below shows the completed menus.

First you will add the Cycle Counts menu option to the replenishment menu.

- In the HighJump One Platform UI, click the **Add** link on the bottom row.

The system displays the Edit/Add Menu window.
Admin Lab Exercise 4: Terminal Menus

The Locale, Menu Level, and Menu Name are all read-only settings based upon the previous page. The Text setting is the text of the menu option that will appear on the replenishment menu.

1. Enter **Cycle Counts** in the Text edit box.

If the menu option will launch a business process, then the Process Object / Sub Menu setting should point to the name of the process object as defined in Advantage Architect. If the menu option will connect to a submenu, then the Process Object / Sub Menu setting should contain an identifier for the submenu. (The identifier will never display on the RF terminal.) In this example, the new menu option connects to a submenu, so you will enter a submenu identifier.

2. Enter **CCSUB** in the Process Object / Sub Menu edit box.

3. Click the Insert button in the Action Bar.

The system displays the modifications in the Replenishment menu page.

Note that the system placed the CCSUB text in the Process column. By default, the system assumes that you want to launch a business process, and it places the new text in the Process column. However, in this example, we want the Cycle Counts menu option to connect to a submenu. In order to switch from a business process to a submenu, you need to convert the menu option.
Click the **To Sub Menu** link for the row where the **Text** column is **Cycle Counts**.

Behind the scenes the system switches the Cycle Counts menu option to a submenu, and then it presents the Edit/Add Menu window. On this screen you will define the first menu option that appears on the Cycle Counts submenu. In this example the first menu option on the submenu reads “CC Check”, and it launches the Cycle Count Check business process.

The Locale, Menu Level, and Menu Name are all read-only settings based upon the previous page. The Text setting is the text of the menu option that will appear on the replenishment menu.

Enter **CC Check** in the **Text** edit box.

Enter **Cycle Count Check** in the **Process Object / Sub Menu** edit box.

Click the **Insert** button in the **Action Bar**.

The system shows the Replenishment submenu. It has converted the Cycle Counts menu option from a process-oriented option to a submenu.

In order to view the option which launches the Cycle Count Check business process (the one you just created), you need to drill into the Cycle Counts submenu.
☐ Click the CCSUB link for the row where the Text column is Cycle Counts.

The system displays the contents of the Cycle Counts submenu. It contains one menu option with a title of “CC Check”, and that menu option launches a process object called “Cycle Count Check”.

☐ Click the Add link on the bottom row.

Once again the system presents the Edit/Add Menu window.

The Locale, Menu Level, and Menu Name are all read-only settings based upon the previous page.

☐ Enter Cycle Count in the Text edit box.

☐ Enter Interleave Dispatcher in the Process Object / Sub Menu edit box.

☐ Click the Insert button in the Action Bar.

The system displays the contents of the Cycle Counts submenu. It now contains two menu options, and each menu option launches a business process.
You have made all of the necessary changes in the HighJump One Platform UI. Next you will test the changes on the terminal. In order to see the change on the terminal you need to navigate away from the current screen and then return to it.

☐ On the Virtual Terminal, press the F1 key to navigate back to the main menu.

☐ Enter 3 (Replenishment) at the Option prompt.

The system displays the Replenishment menu on the terminal with the new Cycle Count submenu.

☐ Enter 4 (Cycle Counts) at the Option prompt.

The system displays the new Cycle Counts submenu.
Enter 1 (CC Check) at the Option prompt.

The system launches the Cycle Count Check business process. Because the cycle count data has not been set up on the training machines, the system displays a No Work Is Available screen.

Press the Enter key.

The system returns to the Cycle Counts submenu.

Enter 2 (Cycle Count) at the Option prompt.

The system launches the Cycle Count business process. In this case, there are some pending cycle count tasks, so the system directs you to begin counting the first location.
Press the F1 key to exit the Cycle Count business process.

The system displays a confirmation screen.

Enter Y at the confirmation screen.

The system returns to the cycle counts menu.
In this activity you have renamed the menu options; you have reordered the menu options; you have removed some menu option; and you have added a submenu with links to business processes. By using these skills in your own environment, you can tailor the menus of the RF terminals to your liking.
Admin Lab Exercise 5: Virtual Terminals and Web Terminals

Introduction

This exercise consists of the following activities:

1. Define a Virtual Terminal in Advantage Commander
2. Define a Database Login for a Virtual Terminal
3. Configure a Virtual Terminal on a Workstation
4. Define a Web Terminal in Advantage Commander
5. Define a Database Login for a Web Terminal
6. Configure a Web Terminal on a Workstation
7. Configure an RF Terminal

Background Information

The Advantage system provides three primary methods for running business processes:

- **RF Terminals** are physical devices on the warehouse floor.
- **Web Terminals** are emulation devices that run in an internet browser.
- **Virtual Terminals** are stand-alone tool emulation devices that run on a desktop.

Before any of these terminals can be used, however, they must be defined and configured in a couple different interfaces. The activities below demonstrate how to define and configure a Virtual Terminal and a Web Terminal in the Advantage system.

Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Add RF, VT, WT Devices menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Define a Virtual Terminal in Advantage Commander

Introduction: A Virtual Terminal is a client application that runs either on a workstation or the application server. The training environment you are using already has Virtual Terminal installed on it. Therefore, the procedures below do not detail how to install the program. For additional information on how to install Virtual Terminal on a workstation see the Advantage Platform Installation Guide.

Situation: The warehouse manager calls you on the phone. He explains that the shipping manager would like the ability to perform business processes in his office. Since the RF coverage was not designed to reach into the shipping office, a physical RF reader is not an option. The two of you decide to meet the requirement by giving the shipping manager access to a Virtual Terminal.

Procedures: Defining a Virtual Terminal consists of four steps.
1. Install the Virtual Terminal program on the client.
2. Define the Virtual Terminal in Advantage Commander.
3. Create a database login for the Virtual Terminal.
4. Configure the Virtual Terminal on the workstation.

The procedures in the activity below and the two activities that follow it demonstrate these four steps.

Step 1: Install the Virtual Terminal on the Client

The Virtual Terminal program has already been installed on the training machines. You have been using it in previous exercises. There is nothing to do for this step on the training machine. If you need to install Virtual Terminal on a workstation in your warehouse, you will need to obtain the HighJump installation disks and run “install.exe”.

Step 2: Define the Virtual Terminal in Advantage Commander

Advantage Commander is a web-based tool which allows you to perform a multitude of administrative functions. In this step, you will use Advantage Commander to inform the HighJump system that there is another terminal which will execute business processes.

- Open the HighJump One Platform UI, if necessary.
- Click the Menu | Advantage Commander | Advantage Commander | Network Devices | Sockets menu.
A socket is a collection of Advantage devices that communicate with the application server through the same Windows port. A socket is different from the device definition. Many devices can be attached to the same socket.

If you have not specified the application server on a previous iteration, then the system will display the Select Application Server or Device Template page instead of a socket-related page.

The HighJump system supports multiple application servers. Each socket (and therefore each device) must be associated with an application server that will manage its communications. You must specify an application server for the socket. All of the application servers are listed on this page. There is an additional option to create a template.

If the system displays the Select Application Server or Device Template screen, then click the Select a Solution Environment on Server [machine name] link.

If you chose an application server, then the system displays the Solution Environment page. Each socket (and therefore each device) must be associated with a single solution environment. In this case, you want the socket (and therefore the associated Virtual Terminal) to be a member of the Warehouse Advantage solution environment.
If the system displays the Select Solution Environment screen, then click the **Work with Solution Environment [WA]** link.

At last, the system displays a list of sockets. The application name and server name are displayed in the page title.

The sample data in the training environment contains one socket definition. You could attach the new Virtual Terminal to the pre-defined socket, and it would operate correctly. However, for the purpose of training, this exercise directs you to create a new socket definition and then attach the new Virtual Terminal definition to the new socket.

Click the **New Socket** button in the **Action Bar**.

The system displays the Edit/Add Device window.
It is common for a warehouse to spread the terminals across multiple socket definitions. In this example, you will create a socket definition intended to manage all of the Virtual Terminals in the system.

- Type `VT_SOCKET` in the **Name** edit box.
- Type `Socket for Virtual Terminals` in the **Description** edit box.
- Type `VTSOCK` in the **ID** edit box.

When the system writes an entry in the log regarding this device, it stores the ID in the log and not the name. In a similar fashion, when you look at the log entry in Advantage Commander, it displays the ID and not the name. Therefore it is important to give the ID a meaningful value, in spite the 6-character limitation.

- Choose **Active** in the **Status** drop-down list.
- Type `4500` in the **Port Number** edit box.
☐ Click the **Insert** button.

The system indicates that the new device has been created.

![Inserting new device succeeded.](image)

<table>
<thead>
<tr>
<th>Solution Name</th>
<th>Server Name</th>
<th>Server Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>HJU-VLAB</td>
<td>HJU-VLAB</td>
<td>Succeeded</td>
</tr>
</tbody>
</table>

☐ Click the **OK** button.

The system displays a list of all sockets defined in the system, including the one you just added.

![Socket Devices [HJU-VLAB::WA]](image)

<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
<th>Name</th>
<th>Description</th>
<th>Devices</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Active</td>
<td>Socket Controller 1</td>
<td></td>
<td>Devices</td>
<td>4400</td>
</tr>
<tr>
<td>2</td>
<td>Active</td>
<td>VT_SOCKET</td>
<td>Socket for Virtual Terminals</td>
<td>Devices</td>
<td>4500</td>
</tr>
</tbody>
</table>

The above steps demonstrated how to create a socket definition. However, the Virtual Terminal in the shipping manager’s office will still not work. The Virtual Terminal has not yet been defined in Advantage Commander. The steps below show how to define a Virtual Terminal and how to attach it to an existing socket definition.

☐ Click the **Devices** link associated with the **VT_SOCKET** socket definition.

The system displays a list of all devices associated with the socket. No devices are attached because you just created the socket.
Child Device(s) of [VT_SOCKET - Port 4500]

1. Click the Add Single Device button in the Action Bar.

2. Choose Advantage Virtual Terminal in the Device Type drop-down list.

3. Click the Add button.

The system displays a default template for a Virtual Terminal definition.

4. Type VT_001 in the Name edit box.

5. Type Virtual Terminal 001 in the Description edit box.

6. Type VT001 in the ID edit box.

   This ID value is comparable to the ID of the socket. When the system writes an entry in the log regarding this device, it stores the ID in the log and not the name.

7. Choose Active in the Status box.


When the Virtual Terminal starts, it launches a process object that has been created in the Advantage Architect tool. A process object is similar to a procedure in other formal programming languages. In this case, the Log-on process object prompts for a user id, a password, and a piece of equipment; it manages menu interactions; and it calls the appropriate business process.
Choose **Do Not AutoStart** in the **Startup** box.

All Virtual Terminals have an associated process manager that manages the business logic (screen layout, variables, data input, database interactions…) for the device. The process manager runs on the application server, and is one of the components that may initiate when the solution environment starts. If the AutoStart option is selected, the engine will automatically initiate the process manager for that device when the solution environment starts. If the Do Not AutoStart option is selected, the engine will not initiate the process manager on solution environment startup. Instead it will initiate the process manager only when it is necessary. In either case, the Virtual Terminal will work correctly. This parameter only dictates at what point the engine starts the process manager.

**Select Device Type**

- **Name**: VT_001
- **Description**: Virtual Terminal 001
- **ID**: VT001
- **Status**: Active
- **Process Object**: Log-on
- **Startup**: Do not AutoStart
- **Log Level**: Errors and Warnings

Click the **Insert** button on the page.

The system indicates that the new device has been created.

**Inserting new device succeeded.**

The following servers have been notified.

<table>
<thead>
<tr>
<th>Solution Name</th>
<th>Server Name</th>
<th>Server Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>HJU-VLAB</td>
<td>HJU-VLAB</td>
<td>Succeeded</td>
</tr>
</tbody>
</table>

Click the **OK** button on the page.

The system displays a list of all devices associated with the given socket, including the one you just created.
At this point the Virtual Terminal has been completely defined in Advantage Commander. However, the Virtual Terminal on the shipping manager’s workstation will still not run properly. Every Virtual Terminal must have its own database login. The next activity describes how to create a database login for the new Virtual Terminal.
Activity 2: Define a Database Login for a Virtual Terminal

**Situation:** This scenario is a continuation of the previous one. See the previous activity for a description of the situation.

**Step 3:** Create a Database Login for the Virtual Terminal.

In the previous activity you defined a Virtual Terminal in Advantage Commander. However, that alone is not enough to make the Virtual Terminal work properly. Every Virtual Terminal must have its own database login in order to access the data in the database. The procedures below demonstrate how to create a database login for the new Virtual Terminal.

Defining a database login for a Virtual Terminal does not apply to DCA. The DCA application manages the database logins differently than WA. It is required in the training environment because it uses the WA application as a basis for learning.

- Choose the Start | All Programs | Microsoft SQL Server 2014 | SQL Server 2014 Management Studio menu.
- Type the **machine name** of the database server in the **Server Name** drop-down list.
- Choose **SQL Server Authentication** in the **Authentication** drop-down list.
- Use the **login.txt** file in the folder on the desktop to populate the **Login** and **Password** boxes.
- Click the **Connect** button.

The system displays the Object Explorer window.
Expand the **Security** node.

Right-click the **Logins** node.

Choose the **New Login** menu.

The system opens a page for defining the settings of the new database login.

Type **VT_001** in the **Login Name** edit box.

The SQL Server Login Name must match the Name of the device in Advantage Commander.
Select the SQL Server Authentication radio button.

Type HJSPASS#1 in the Password edit box.

Type HJSPASS#1 in the Confirm Password edit box.

Even though every device logs into the database with a distinct login name, the password for each login name is the same. When a Virtual Terminal needs to connect to the database, the HighJump system uses the Name parameter for the device from Advantage Commander and the DefaultSQLServerPassword parameter from the Program Data \ HighJump Software \ Control \ Control1 \ CONTROL.INI file. The Password parameter in the SQL Server environment must match the DefaultSQLServerPassword parameter in the CONTROL.INI file.
☐ Uncheck the **Enforce password expiration** check box.

☐ Choose **AAD** in the **Default Database** drop-down list.

☐ Click the **User Mapping** node near the upper left corner of the window.

☐ Check the **Map** checkbox next to the **AAD** database in the upper pane. The system automatically populates the User column with the name of the device.

☐ Check the **Role Membership** checkbox next to **AAD_USER** in the lower pane.

☐ Check the **Role Membership** checkbox next to **WA_USER** in the lower pane.

☐ Check the **Map** checkbox next to the **ADV** database in the upper pane. The system automatically populates the User column with the name of the device.
Check the Role Membership checkbox next to AAD_USER in the lower pane.

Check the Role Membership checkbox next to ADV_USER in the lower pane.

Check the Role Membership checkbox next to WA_USER in the lower pane.

Click the OK button.

Expand the Logins node in the left panel.

Scroll down to the bottom of the list.

The system displays the new login along with all of the other logins included in the base solution.

At this point the Virtual Terminal login has been completely defined in the database. However, the Virtual Terminal on the shipping manager’s workstation will still not run properly. Every Virtual Terminal must also be configured on the workstation. The next activity describes how to configure the new Virtual Terminal on the workstation.
Activity 3: Configure a Virtual Terminal on a Workstation

Situation: This scenario is a continuation of the previous two. See the previous activities for a description of the situation.

Step 4: Configure the Virtual Terminal on the Workstation.

In the previous activities you defined a Virtual Terminal in Advantage Commander and you created an associated database login in the database. However, that alone is not enough to make the Virtual Terminal work properly. Every Virtual Terminal must also be configured at the workstation.

☐ Start the engine for the WA solution environment, if necessary.

☐ Start the Virtual Terminal, if necessary.

☐ Press the F1 key to navigate backwards to the USER ID prompt, if necessary.

This Virtual Terminal is already installed on the workstation and it is working properly. However, this Virtual Terminal is not VT_001 (the device you defined in the previous activities.) This device was installed and configured on the machine during the installation of the Advantage Platform.

The Advantage Platform limits the number of Virtual Terminals that can run on a single machine. It only supports one Virtual Terminal per workstation. Rather than installing another instance of Virtual Terminal, the procedures below describe how to change the configuration of the existing Virtual Terminal so that it uses the device definition of VT_001.

☐ Right-click anywhere on the face of the Virtual Terminal.

☐ Choose the Configuration menu.

The system displays the Virtual Terminal Configuration window.

☐ Type VT_001 in the Device Name edit box.
The Device Name of the Virtual Terminal configuration must match the Name of the device definition in Advantage Commander.

- Type the **machine name** of the application server in the **Host Server Name** edit box.

- Type **4500** in the **Host Port Number** edit box.

The Host Port Number of the Virtual Terminal configuration must match the Port Number of the associated socket definition in Advantage Commander.
☐ Click the OK button.

The system displays the welcome dialog and USER ID prompt of the Virtual Terminal.

The activities above directed you to manage four different aspects of the Virtual Terminal definition:
1. Install the Virtual Terminal program on the client.
2. Define the Virtual Terminal in Advantage Commander.
3. Create a database login for the device.
4. Configure the Virtual Terminal on the workstation.

At this point, the definition phase is complete. The final step in this process is to validate that the Virtual Terminal works as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

The test cases below do not detail every keystroke. You need to use the knowledge you gained in previous activities to test these scenarios.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Perform the steps from the above four activities</td>
<td>The Virtual Terminal prompts for a USER ID</td>
</tr>
<tr>
<td></td>
<td>Enter a valid user at the USER ID dialog.</td>
<td>System advances to PASSWORD dialog.</td>
</tr>
<tr>
<td></td>
<td>Enter the user’s password at the PASSWORD dialog.</td>
<td>System advances to the EQUIPMENT / ZONE dialog.</td>
</tr>
<tr>
<td></td>
<td>Enter a valid forklift at the EQUIPMENT / ZONE dialog.</td>
<td>System advances to the main menu.</td>
</tr>
</tbody>
</table>
After confirming the new Virtual Terminal works as expected, follow the steps below to restore the settings. (The original Virtual Terminals will be used in upcoming exercises.)

☐ Press the **F1 key** to navigate backwards to the **USER ID** prompt on the **Virtual Terminal**, if necessary.

☐ Right-click anywhere on the face of the **Virtual Terminal**.

☐ Choose the **Configuration** menu.

☐ Type **RADIO_01** in the **Device Name** edit box.

☐ Enter the **machine name** of the application in the **Host Server Name** edit box.

☐ Type **4400** in the **Host Port Number** edit box.

The settings should look like the following.

☐ Click the **OK** button.
Activity 4: Define a Web Terminal in Advantage Commander

Introduction: A Web Terminal allows an individual to run business processes (just like a Virtual Terminal or an RF reader) through an Internet Explorer session. One advantage of the Web Terminal is that it does not require a client application other than Internet Explorer.

Situation: The warehouse manager calls you on the phone. He explains that there are several additional managers and supervisors scattered throughout the warehouse who would like the ability to perform business processes in their various offices. Since the RF coverage was not designed to reach all of the offices, physical RF readers are not an option. Additionally, you don't want to take the time to install a client application on each of the individual workstations. The two of you decide to meet the requirement by giving the other managers and supervisors the ability to perform business processes through a Web Terminal.

Procedures: Defining a Web Terminal consists of four steps.
1. Verify Internet Explorer exists on the client.
2. Define the Web Terminal in Advantage Commander.
3. Create a database login for the Web Terminal.
4. Configure the Web Terminal on the workstation.

The procedures in the activity below and the two activities that follow it demonstrate these four steps.

Step 1: Verify Internet Explorer exists on the Client

Unlike the Virtual Terminal, the Web Terminal does not use a client program – other than Internet Explorer. Since the training machine already contains Internet Explorer, there is nothing to do for this step. If you need to implement Web Terminal on a workstation in your warehouse, you need only to validate that Internet Explorer exists on the client workstation.

Step 2: Define the Web Terminal in Advantage Commander

Advantage Commander is a web-based tool which allows you to perform a multitude of administrative functions. In this step, you will use Advantage Commander to inform the HighJump system that there is another terminal which will execute business processes.

- Open the HighJump One Platform UI, if necessary.
- Click the Menu | Advantage Commander | Advantage Commander | Network Devices | Sockets menu.
A socket is a collection of Advantage devices that communicate with the application server through the same Windows port. A socket is different from the device definition. Many devices can be attached to the same socket.

If you have not specified the application server on a previous iteration, then the system will display the Select Application Server or Device Template page instead of a socket-related page.

The HighJump system supports multiple application servers. Each socket (and therefore each device) must be associated with an application server that will manage its communications. You must specify an application server for the socket. All of the application servers are listed on this page. There is an additional option to create a template.

If the system displays the Select Application Server or Device Template screen, then click the Select a Solution Environment on Server [machine name] link.

If you chose an application server, then the system displays the Solution Environment page. Each socket (and therefore each device) must be associated with a single solution environment. In this case, you want the socket (and therefore the associated Virtual Terminal) to be a member of the Warehouse Advantage solution environment. If the system did not present the Application Server page, then neither will it present the Solution Environment page.
If the system displays the Select Solution Environment screen, then click the Work with Solution Environment [WA] link.

Select Solution Environment

<table>
<thead>
<tr>
<th>#</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Work with Solution Environment [AFA]</td>
</tr>
<tr>
<td>2</td>
<td>Work with Solution Environment [AdvLinkFlatFile]</td>
</tr>
<tr>
<td>3</td>
<td>Work with Solution Environment [WA]</td>
</tr>
<tr>
<td>4</td>
<td>Work with Solution Environment [WAPROCESSORS]</td>
</tr>
</tbody>
</table>

At last, the system displays a list of sockets. The application name and solution environment are displayed in the page title.

Socket Devices [HJU-VLAB::WA]

<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
<th>Name</th>
<th>Description</th>
<th>Devices</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Active</td>
<td>Socket Controller 1</td>
<td></td>
<td>Devices</td>
<td>4400</td>
</tr>
<tr>
<td>2</td>
<td>Active</td>
<td>VT_SOCKET</td>
<td>Socket for Virtual Terminals</td>
<td>Devices</td>
<td>4500</td>
</tr>
</tbody>
</table>

You could attach the new Web Terminal to one of the existing sockets, and it would operate correctly. However, for the purpose of training, this exercise directs you to create a new socket definition and then attach the new Web Terminal definition to the new socket.

Click the New Socket button in the Action Bar.

The system displays the Edit / Add Device page.
It is common for a warehouse to spread the terminals across multiple socket definitions. In this example, you will create a socket intended to manage all of the Web Terminals in the system.

☐ Type **WT_SOCKET** in the **Name** edit box.

☐ Type **Socket for Web Terminals** in the **Description** edit box.

When the system writes an entry in the log regarding this device, it stores the ID in the log and not the name. In a similar fashion, when you look at the log entry in Advantage Commander, it displays the ID and not the name. Therefore it is important to give the ID a meaningful value, in spite the 6-character limitation.

☐ Type **WT_sock** in the **ID** edit box.

☐ Choose **Active** in the **Status** drop-down list.

☐ Type **4600** in the **Port Number** edit box.

The completed socket settings should look like the following graphic.
Click the **Insert** button.

The system indicates that the new device has been created.

**Inserting new device succeeded.**

The following servers have been notified.

<table>
<thead>
<tr>
<th>Solution Name</th>
<th>Server Name</th>
<th>Server Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>HJU-VLAB</td>
<td>HJU-VLAB</td>
<td>Succeeded</td>
</tr>
</tbody>
</table>

Click the **OK** button.

The system displays a list of all sockets defined in the system, including the one you just added.
The above steps demonstrated how to create a socket definition. However, the Web Terminal in the shipping manager’s office will still not work. The Web Terminal has not yet been defined in Advantage Commander. The steps below show how to define a Web Terminal and how to attach it to an existing socket definition.

☐ Click the **Devices** link associated with the **WT_SOCKET** socket definition.

The system displays a list of all devices associated with the socket. No devices are attached because you just created the socket.

☐ Click the **Add Single Device** button in the **Action Bar**.

The system displays the Select Device Type page.

☐ Choose **Advantage Web Terminal** in the **Device Type** drop-down list.

☐ Click the **Add** button.

The system displays the Edit/Add Device page.
☐ Type **WT_001** in the **Name** edit box.

☐ Type **Web Terminal 001** in the **Description** edit box.

This ID value is comparable to the ID of the socket. When the system writes an entry in the log regarding this device, it stores the ID in the log and not the name.

☐ Type **WT001** in the **ID** edit box.

☐ Choose **Active** in the **Status** box.

When the Web Terminal starts, it launches a process object that has been created in the Advantage Architect tool. A process object is similar to a procedure in other formal programming languages. In this case, the Log-on process object prompts for a user id, a password, and a piece of equipment; it manages menu interactions; and it calls the appropriate business process.

☐ Choose **Log-on** in the **Process Object** drop-down list.

The remainder of the settings can remain as their defaulted values. The completed Web Terminal settings should look like the following graphic.
Admin Lab Exercise 5:
Virtual Terminals and Web Terminals

☐ Click the **Insert** button.

The system indicates that the new device has been created.

**Inserting new device succeeded.**

<table>
<thead>
<tr>
<th>Solution Name</th>
<th>Server Name</th>
<th>Server Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>HJU-VLAB</td>
<td>HJU-VLAB</td>
<td>Succeeded</td>
</tr>
</tbody>
</table>

☐ Click the **OK** button.

The system displays a list of all devices associated with the given socket, including the one you just created.

**Child Device(s) of [WT_SOCKET - Port 4600]**

<table>
<thead>
<tr>
<th>#</th>
<th>Status</th>
<th>Name</th>
<th>ID</th>
<th>Description</th>
<th>Type</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Active</td>
<td>WT_001</td>
<td>WT001</td>
<td>Web Terminal 001</td>
<td>Web Browser Terminal</td>
<td>HJU-VLAB</td>
</tr>
</tbody>
</table>

At this point the Web Terminal has been completely defined in Advantage Commander. However, the Web Terminal on the shipping manager’s workstation will still not run properly. Every Web Terminal must
have its own database login. The next activity describes how to create a database login for the new Web Terminal.
Activity 5: Define a Database Login for a Web Terminal

Situation: This scenario is a continuation of the previous one. See the previous activity for a description of the situation.

Step 3: Create a Database Login for the Web Terminal.

In the previous activity you defined a Web Terminal in Advantage Commander. However, that alone is not enough to make the Web Terminal work properly. Every Web Terminal must have its own database login in order to access the data in the database. The procedures below demonstrate how to create a database login for the new Web Terminal.

Defining a database login for a Virtual Terminal does not apply to DCA. The DCA application manages the database logins differently than WA. It is required in the training environment because it uses the WA application as a basis for learning.

☐ Choose the Start | All Programs | Microsoft SQL Server 2014 | SQL Server 2014 Management Studio menu.

☐ Type the machine name of the database server in the Server Name drop-down list.

☐ Choose SQL Server Authentication in the Authentication drop-down list.

☐ Use the login.txt file in the folder on the desktop to populate the Login and Password boxes.

The system displays the Object Explorer window.
Expand the **Security** node.

Right-click the **Logins** node.

Choose the **New Login** menu.

The system opens a page for defining the settings of the new database login.

- **Type WT_001** in the **Login Name** edit box.

The SQL Server Login Name must match the Name of the device in Advantage Commander.

- **Select the **SQL Server Authentication** radio button.**
- **Type HJSPASS#1** in the **Password** edit box.
- **Type HJSPASS#1** in the **Confirm Password** edit box.

Even though every device logs into the database with a distinct login name, the password for each login name is the same. When a Web Terminal needs to connect to the database, the HighJump system uses the Name parameter for the device from Advantage Commander and the DefaultSQLServerPassword.
parameter from the \HIGHJUMP SOFTWARE\CONTROL\CONTROL1\CONTROL.INI file. The Password parameter in the SQL Server environment must match the DefaultSQLServerPassword parameter in the CONTROL.INI file.

- Uncheck the **Enforce password expiration** check box.

- Choose **AAD** in the **Default Database** drop-down list.

- Click the **User Mapping** node near the upper left corner of the window.

- Check the **Map** checkbox next to the **AAD** database in the upper pane.

The system automatically populates the User column with the name of the device.
Admin Lab Exercise 5:
Virtual Terminals and Web Terminals

- Check the Role Membership checkbox next to AAD_USER in the lower pane.
- Check the Role Membership checkbox next to WA_USER in the lower pane.

- Check the Map checkbox next to the ADV database in the upper pane.
  The system automatically populates the User column with the name of the device.

- Check the Role Membership checkbox next to AAD_USER in the lower pane.
- Check the Role Membership checkbox next to ADV_USER in the lower pane.
- Check the Role Membership checkbox next to WA_USER in the lower pane.
☐ Click the **OK** button.

☐ Expand the **Logins** node in the left panel.

☐ Scroll down to the **bottom** of the list.

The system displays the new login along with all of the other logins included in the base solution.

At this point the Web Terminal login has been completely defined in the database. However, the Web Terminal on the shipping manager’s workstation will still not run properly. Every Web Terminal must also be configured on the workstation. The next activity describes how to configure the new Web Terminal on the workstation.
Activity 6: Configure a Web Terminal on a Workstation

Situation: This scenario is a continuation of the previous two. See the previous activities for a description of the situation.

Step 4: Configure the Web Terminal on the Workstation

In the previous activity you defined a Web Terminal in Advantage Commander. However, that alone is

In the previous activities you defined a Virtual Terminal in Advantage Commander and you created an associated database login in the database. However, that alone is not enough to make the Virtual Terminal work properly. Every Virtual Terminal must also be configured at the workstation.

☐ Start the engine for the WA solution environment, if necessary.

☐ Open Internet Explorer.

☐ Type http://<web server name>/webtrmgw/webtrmgw.dll in the address bar.

The <web server name> reference above is not literal text. It should be replaced with the machine name of the web server.

The system displays a dialog that is similar to the configuration dialog for Virtual Terminals.

☐ Type the machine name of the application server in the Advantage Application Server Name edit box.

☐ Type 4600 in the Port Number edit box.

Just like the Virtual Terminal, the Port Number of the Web Terminal configuration must match the Port Number of the associated socket definition in Advantage Commander.
Type **WT_001** in the **Device Name** edit box.

Just like the Virtual Terminal, the Device Name of the Web Terminal configuration must match the Name of the device definition in Advantage Commander.
Click the **Submit** button.

The system displays the welcome dialog and USER ID prompt of the Web Terminal. Apart from the function keys on the bottom of the display, this terminal looks very similar to a Virtual Terminal a physical RF terminal.

![Web Terminal Welcome Dialog](image)

The activities above directed you to manage three different aspects of the Web Terminal definition:

- The Web Terminal definition in Advantage Commander.
- The database login for the device in the database.
- The Web Terminal configuration on the workstation.

At this point, the definition phase is complete. The final step in this process is to validate that the Web Terminal works as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

The test cases below do not detail every keystroke. You need to use the knowledge you gained in previous activities to test these scenarios.
## Admin Lab Exercise 5:
### Virtual Terminals and Web Terminals

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Perform the steps from the above three activities.</td>
<td>The Web Terminal prompts for a USER ID.</td>
</tr>
<tr>
<td></td>
<td>Enter a valid user at the USER ID dialog.</td>
<td>System advances to PASSWORD dialog.</td>
</tr>
<tr>
<td></td>
<td>Enter the user’s password at the PASSWORD dialog.</td>
<td>System advances to the EQUIPMENT / ZONE dialog.</td>
</tr>
<tr>
<td></td>
<td>Enter a valid forklift at the EQUIPMENT / ZONE dialog.</td>
<td>System advances to the main menu.</td>
</tr>
</tbody>
</table>
Activity 7: Configure an RF Terminal

In the previous activities you configured a Virtual Terminal and a Web Terminal. These emulation devices are useful in testing, training, and limited office-based production activities. However, the terminal you will probably use the most in the warehouse is a physical RF Terminal which you will hold in your hand; wear on your wrist; or mount on a forklift.

This training class does not provide you with a physical RF Terminal. So instead of giving exact procedures on how to define an RF Terminal (which you wouldn’t be able to test), this manual gives a high-level description of the process and explains how it differs from a Virtual Terminal.

Configuring an RF Terminal involves three steps which are somewhat similar to the ones you used for configuring a Virtual Terminal. The three steps are listed below:

Step 1. Configure the Network Settings and the Telnet Client Settings on the RF Terminal
Step 2. Define the RF Terminal in Advantage Commander.
Step 3. Create a database login for the RF Terminal.

Step 1: Configure the Network Settings and the Telnet Client Settings on the RF Terminal

The physical RF Terminal communicates with the HighJump application server across the RF network. The first step of the configuration process is to configure the network settings on the RF Terminal so that it will properly communicate with the RF network. When you configure these network settings you must provide some mechanism for identifying the RF Terminal. This setting will take the form of either a unique IP address or some other unique identifier.

The HighJump application server and the RF Terminals communicate with one another via a communication protocol called telnet. HighJump uses a solution from Georgia SoftWorks to provide the telnet communication software and the associated telnet administrative tools. Every physical RF Terminal runs a telnet client session which communicates with the Georgia SoftWorks server components. The next step of the configuration process is to configure the telnet client settings on the RF Terminal to communicate with Georgia SoftWorks.

The actual procedures for defining the network settings and the telnet client settings will differ depending upon the brand and model of the RF Terminal. Hardware acquired through HighJump may already be preconfigured as part of the purchase agreement. Consider this point before overriding or resetting the devices settings.

Step 2: Define the RF Terminal in Advantage Commander

When you defined a Virtual Terminal in Advantage Commander, you created a socket and then you attached multiple Virtual Terminals (devices) to the socket. The same principles apply for an RF Terminal. You still need a socket, and you still attach the RF Terminals to a socket. However, there are two key differences when defining an RF Terminal in Advantage Commander.
The first difference revolves around the Device Type setting. When you click the Add Single Device button, the system displays a window asking you to indicate the type of device you want to add. For RF Terminals, you must select the Advantage Telnet Terminal menu option in the Device Type drop-down list. (Telnet is the communication protocol between the RF Terminal and the application server.)

The second difference revolves around a device-specific setting. When you, or HighJump, configured the network settings on the RF Terminal, you provided a unique IP address or some other unique identifier to identify the device. When you define the RF Terminal in Advantage Commander you must account for this setting. If the unique identifier was an explicit IP Address, then you must enter this value in the IP Address edit box on the Edit/Add Device screen. If the unique identifier was something other than an explicit IP Address, then you must enter this value in the Unique Identifier edit box on the Edit/Add Device screen. The other required settings on the Edit/Add Device page were discussed in the section on configuring Virtual Terminals.
**Step 3**: Create a database login for the RF Terminal.

Creating a database login for an RF Terminal is identical to the process used for creating a database login for a Virtual Terminal.
Admin Lab Exercise 6: Advantage Platform Log

Introduction

This exercise consists of the following activities:

1. View the Log Reports
2. Export the Log with the HighJump One Platform User Interface (UI) Export Feature
3. Export the Log with the Log Export Utility

Background Information

The Advantage Platform log is a collection of messages stored in a database table related to the HighJump system. Some of the messages in the logs are informational. For example, “Starting the Advantage Engine.” Some of the messages indicate failures. For example, “Database Failure.” Others provide detailed application tracing information. For example, “Updated field [Quantity] to [73].”

One of the common tasks of an administrator is to determine the root cause of problems. The logs are an excellent source of information for identifying problems in the HighJump system. They should be one of the first places you look when you are unable to establish the root cause of a symptom. Advantage Commander is the tool that provides the ability to view the log messages.

The log messages are broken down into two broad categories: application messages and system messages. The chart below gives a brief description of the two types of messages.

<table>
<thead>
<tr>
<th>Type</th>
<th>Notes</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Log Message</td>
<td>Related to business logic (workflows)</td>
<td>SQL Execute Failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cleared 0 location user counts</td>
</tr>
<tr>
<td>System Log Message</td>
<td>Non-business logic content. It includes</td>
<td>Started Advantage Workflow Engine</td>
</tr>
<tr>
<td></td>
<td>starting and stopping managers, as well as</td>
<td>Checking Software License</td>
</tr>
<tr>
<td></td>
<td>communications between components.</td>
<td></td>
</tr>
</tbody>
</table>

Often times the administrator is not the only individual interested in reading the logs. Sometimes outside parties, like the HighJump support team, have a vested interest in the content of the logs. In these cases, it is beneficial to export the logs out of the database tables into a portable format. Then you can deliver the exported logs to the interested parties.
HighJump provides two methods for exporting the logs out of the database tables. The first method uses the HighJump One Platform UI, and it creates a Microsoft Excel document. The second method uses a stand-alone tool called the Log Export Utility, and it generates a XML file. Other differences between the two methods are discussed in the individual activities.

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the Log Reports menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: View the Log Reports in Advantage Commander

Situation: One of the warehouse supervisors calls you on the phone. He explains that one of the RF readers is displaying an error that reads, “Database error has occurred. Notify your system manager.” Regardless of what the supervisor does on the RF reader, it always returns to this screen. You suspect that the problem is at the application layer, but you would like to do a little bit of research before you hand it over to the application development team.

Procedures: The procedures below describe how to view the log for information related to the errors. However, they do not recreate the “Database error” message mentioned above.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Advantage Commander | Advantage Commander | Log | Summary Log Report menu.

The system displays the Summary Log Search Criteria page.

By default, the system sets the filter criteria to all log messages within the last twenty-four hours. If you want to narrow the result set, you can change the filter criteria here. For example, if you were only interested in application log messages, you would choose Application Messages in the Message Types drop down list. For the purposes of this exercise, the defaulted values are sufficient.

☐ Click the Submit button in the Action Bar.

The system displays the Log Messages Summary View page.
Use the horizontal scroll bar and the navigation buttons near the top to browse the messages.

Note the type of messages that are logged here. Many of the messages in the training environment are related to starting and stopping various managers. The system logs these messages when you start and stop solution environments. If all elements of a production environment are operating smoothly, then the number of log messages in the log will be relatively small. As you become more familiar with the log messages, you will begin to recognize some of the more common messages.

Also note, the number of attributes displayed for each log message. This page displays relatively few attributes. While the information displayed in this summary view is most likely adequate to meet your needs, Advantage Commander also provides a second view of the same messages. The second view presents the same messages, but with many more attributes for each message. The procedures below demonstrate how to view this detailed version of the log messages.

Click the Menu | Advantage Commander | Advantage Commander | Log | Detailed Log Report menu.

The system displays the Detailed Log Search Criteria page.
Apart from the title, this page looks and operates in an identical manner to the earlier page. By default, the system sets the filter criteria to all log messages within the last twenty-four hours. If you want to narrow the result set, you can change the filter criteria here. For the purposes of this exercise, the defaulted values are sufficient.

☐ Click the Submit button in the Action Bar.

The system displays the Log Messages Detailed View page.

<table>
<thead>
<tr>
<th>#</th>
<th>Date/Time Logged</th>
<th>IPC Address</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10/6/2016 3:15:34 PM</td>
<td>SMGR</td>
<td>Log Maintenance configuration parameters</td>
</tr>
<tr>
<td>3</td>
<td>10/6/2016 3:14:56 PM</td>
<td>SMGR</td>
<td>Event Queue Maintenance started [dbo.usp]</td>
</tr>
</tbody>
</table>

☐ Use the horizontal scroll bar and the navigation buttons near the top to browse the messages. Note that the page displays the same messages as the previous iteration. However, this page gives many more details associated with each message.
Activity 2: Export the Log with the HighJump One Platform User Interface (UI) Export Feature

**Background:** The HighJump system provides two methods for exporting the log information. The first method uses a feature of the HighJump One Platform UI. The second method uses a stand-alone tool, and it is addressed in the next activity.

The HighJump One Platform UI feature is not exclusive to log messages. Rather, it is a feature that applies to all HighJump One Platform UI report pages. (A report page is a screen that displays a set of data in a grid-type format.) This export method produces a Microsoft Excel document that contains the entire result set.

The HighJump One Platform UI method allows you to easily export the result set while viewing the data on the report page. Additionally, it allows you to filter the data before the export. This may or may not be advantageous depending upon the situation.

**Situation:** One of the warehouse supervisors calls you on the phone. He reports that every day in the last week the RF readers on the receiving dock have taken a serious performance hit around 3:00 in the afternoon for about 10 minutes. The receivers often times wait up to 15 seconds between screens during this timeframe. You have reviewed the log in Advantage Commander and you see some log messages that may be related to the issue at hand. However, you don’t know how to interpret them. The log messages seem to point towards the network. Since you are not versed in networks, you would like the network administrator to review the logs. Rather than instructing the network administrator on how to access the logs in Advantage Commander, you decide that it is simply easier to export the log to an Excel format and then deliver that document to him.

**Procedures:** The procedures below demonstrate how to export the messages to a Microsoft Excel document. However, they do not produce the log messages related to the performance problems mentioned in the narrative above.

- Open the HighJump One Platform UI, if necessary.
- Click the Menu | Advantage Commander | Advantage Commander | Log | Summary Log Report menu.

The system displays the Summary Log Search Criteria page.

```
Summary Log Search Criteria

Start Date and Time: 10/5/2016 3:21 PM
End Date and Time: 10/6/2016 3:21 PM
```
☐ Type the **date from seven days ago** in the **Start Date and Time** edit box.

By changing the start date, the system will return multiple sets of the log messages for the performance problems. The network administrator may be able to determine a pattern by looking at multiple sets of messages.

☐ Click the **Submit** button in the **Action Bar**.

The system displays the Log Messages Summary View page.

<table>
<thead>
<tr>
<th>#</th>
<th>Date/Time Logged</th>
<th>IPC Address</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10/6/2016 3:20:35 PM</td>
<td>SMGR</td>
<td>Log Maintenance configuration parameters -</td>
</tr>
<tr>
<td>3</td>
<td>10/6/2016 3:15:34 PM</td>
<td>SMGR</td>
<td>Log Maintenance configuration parameters -</td>
</tr>
</tbody>
</table>

☐ Click the **Export** button in the **Action Bar**.

The system displays the open/save file prompt at the bottom of the page.

☐ Click the **Save** button.

The system displays the download completed prompt at the bottom of the page.

☐ Click the **Open Folder** button.

The system creates a file on the Downloads folder called Log_Message_Summary_View.xls.
Double-click the **Log_Message_Summary_View** file in the Downloads folder.

The system displays the contents of the Excel file.

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Date/Time Logged</strong></td>
</tr>
<tr>
<td>2</td>
<td>10/15/2015 8:40:06 AM SMGR</td>
</tr>
<tr>
<td>3</td>
<td>10/15/2015 8:45:06 AM SMGR</td>
</tr>
<tr>
<td>4</td>
<td>10/15/2015 8:41:16 AM SMGR</td>
</tr>
<tr>
<td>5</td>
<td>10/15/2015 8:41:16 AM SMGR</td>
</tr>
<tr>
<td>6</td>
<td>10/15/2015 8:41:16 AM SMGR</td>
</tr>
</tbody>
</table>
```

**Details**

- Log Maintenance configuration parameters - LogPurgeMaxin
- Starting Log Maintenance.
- Event Queue Maintenance started [dbo.usp_AdvDatabaseMaint]
- Message Bus Maintenance started [dbo.usp_AdvDatabaseMaint]
- Event Queue Maintenance configuration parameters - Event
Activity 3: Export the Log with the Log Export Utility

Background: The HighJump system provides two methods for exporting the log information. The first method uses a feature of the HighJump One Platform UI, and it was demonstrated in the previous activity. The second method uses a stand-alone tool called the Log Export Utility to perform the export. The Log Export Utility is demonstrated in this activity.

The Log Export Utility produces either a Visual FoxPro file or a tab-delimited file. The recipient of these files can then review the file in Visual FoxPro or in a spreadsheet application (like Excel) respectively.

If you intend to export significant amounts of data, then the Log Export Utility may be a better option. Unlike the Advantage Commander solution, the Log Export Utility does not allow the user to filter the data prior to the export. The Log Export Utility exports all of the log messages in the log table.

Situation: This scenario builds upon the scenario from the previous activity. The network administrator has reviewed the log exported from Advantage Commander. He sees some unusual log activity during the noted times. However, he is not convinced that the logs point towards the network. He suggests contacting the HighJump Worldwide Support team for additional assistance. You call the support team, and they request the log messages from your system. They request that you provide the data using the Log Export Utility because that will give them a more complete picture of the log.

Procedures: The procedures below demonstrate how to export the log messages into a tab-delimited format using the Log Export Utility. This activity focuses on the export side, and it does not address the activities the recipient must take in order to view the log messages.

Choose the Start | All Programs | HighJump Software | Log Export Utility menu.

The system displays the Log Export Utility tool.
☐ Select the **Tab delimited** radio button.

The system defaults the export path and the resulting file name. There is no need to change this value. Additionally, the system provides the ability to compress the file after the export. You will not change this value either.

```
Export Path: D:\Program Data\HighJump\Software\LogMessages\LogMsg.csv
☐ Compress results using zip
```

At this point, you have provided all of the information necessary to create an exported log file. It is now time to perform the actual export.

☐ Click the **Export** button.

The system dumps the HighJump log to a file on the hard drive. When the process is finished, the system writes a message to the bottom portion of the utility which indicates how many records it exported.

```
Export Status:
Starting Export Process ...
Approximately 18,000 log messages will be exported.
Selecting log messages to be exported. Please wait.
Select completed.
Starting to export log messages.
10,000 log messages exported.
Exported 18,116 log messages.
Export Completed.
```

☐ Choose the **File | Exit** menu.

**Test Cases:** The final step in this process is to validate that the system exported the data as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome.

If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
</table>

---

**Admin Lab Exercise 6:**  
**Advantage Platform Log**
| Perform the above procedures. | System creates a LogMsg.csv file in the D:\Program Data\HighJump Software\LogMessages directory. |

**Follow Up:** At this point, you can zip the file and send it to the HighJump Worldwide Support team. The Support team will import the file into one of their local database. Then they will review the log messages and provide you with some direction based upon their analysis.
Admin Lab Exercise 7:
Status Consoles

Introduction

This exercise consists of the following activities:

1. View a Log Message in a Status Console
2. Change the Size of the Status Console Buffer
3. Copy Messages Out of the Status Consoles

Background Information

In the previous lab exercises, you learned how to view the log messages through Advantage Commander. You also learned the difference between an application log message and a system log message.

The status consoles provide another method for viewing log messages. A status console is a window on the application server that displays the log messages as they are being written to the log. Essentially, the status consoles provide a quick way of performing a high-level health check on the HighJump system without having to navigate through the web pages of Advantage Commander.

There are two different status consoles: the application status console and the system status console. The application status console displays application log messages as they are being written to the log. Likewise, the system status console displays system log messages as they are being written to the log.

When you viewed the log messages in Advantage Commander, you had access to the entire history of messages. You could also apply a filter against the messages to narrow the data set to only the relevant data. The status consoles do not operate in this manner. They only display the most recent set of messages written to the log. Additionally, the status consoles can only display a fixed number of messages. When that limit is reached, the oldest entry in the console is removed before a new one is added.

While not required, it is a good practice to keep the consoles open any time the engine is running. If the log message is a recent one, it may be easier to find the message in the status console than the web pages of Advantage Commander.
Before performing this exercise, we recommend you use the **Troubleshooting Advantage System** Pre-Work link and review the related video demonstrations under the **Status Consoles** menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: View a Log Message in a Status Console

**Situation:** Tomorrow is the go-live date for your HighJump system. In preparation for that significant event, you want to follow the best practice noted above and open the application status console and the system status console. And you want to validate that they display the messages correctly.

**Procedures:** The procedures below demonstrate how to open the status consoles and view the log messages through them.

- **Step 1:** Shutdown the **engine** for all solution environments.

  Shutting down the engine is not required in order to start the consoles. It is included here so that you can view the log messages when the engine starts.

- **Step 2:** Choose the **Start | All Programs | HighJump Software | Application Status Console** menu.

  The system opens the Application Status Console. The engine is not running, so the window is empty.

- **Step 3:** Choose the **Start | All Programs | HighJump Software | System Status Console** menu.

  The system opens the System Status Console. The engine is not running, so the window is empty.

- **Step 4:** Start the **engine** for all solution environments.

  The system displays messages in both of the status consoles.
Admin Lab Exercise 7:
Status Consoles
Activity 2: Change the Size of the Status Console Buffer

Background: In the previous activity, you used the status consoles to view the log messages that were generated when the engine started. However, if you review the messages in the consoles, you will note that they are difficult to read because many of the messages span multiple lines.

Situation: You have opened the status consoles in preparation for tomorrow’s go-live event. In order to be in the best position to react to any errors, you decide to make the log messages in the status consoles easier to read.

Procedures: The procedures below demonstrate how to prevent log messages from spanning multiple lines in the status consoles.

☐ Open the Application Status Console, if necessary.

☐ Right-click anywhere on the title bar of the Application Status Console window.

☐ Choose the Properties menu.

The system displays the Properties window for the Application Status Console.

☐ Click the Layout tab.

The system displays the layout settings.

The Screen Buffer Size and the Window Size sections of the Layout tab are both related to the display of the console. The Window Size section relates to the dimensions of the window, but it does not impact the wrapping of the actual text. The Screen Buffer Size section outlines the invisible border in which the
text is written. If a given message exceeds the width of the screen buffer size, then the message wraps to the next line.

- Enter 500 in the Width box of the Screen Buffer Size section.

- Click the OK button.

The system does not change the size of the window. Nor does it change the wrapping behavior of the text. The change you made above does not apply to existing messages. It only applies to the new messages.

- Repeat the above steps for the System Status Console window.

**Test Cases:** The final step in the process is to validate that the change worked as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shutdown the engine. Then start the engine for all solution environments</td>
<td>Both consoles display additional log messages. Each log message is displayed on a single line.</td>
</tr>
</tbody>
</table>

The data in the consoles looks similar to the following.
Activity 3: Copy Messages Out of the Status Consoles

Background: It is important to be able to see the messages in the status consoles. However, there may be times when you want to send some of the messages to another individual for further analysis. In a previous exercise, you learned how to export the log messages through Advantage Commander, and you also learned how to export the log messages with the Log Export Utility. If you were only interested in one or two log messages, you could also simply copy them out of the Advantage Commander web pages and paste them into another application. The status consoles provide another method for extracting log messages. The export feature of Advantage Commander and the Log Export Utility work well if you want to extract a multitude of messages. Because of the limited buffer, the status console method works best when extracting a small number of messages.

Situation: The status consoles have constantly been open on the application server. One afternoon, one of the warehouse supervisors calls you on the phone and indicates that several of the RF terminals display a database failure message during the receiving process. The users can usually navigate around the problem by pressing the enter key. You look at the application status console and you see several messages that appear to be related to the problem. However, you don’t know how to interpret the data. You plan to contact the HighJump support team via e-mail for some additional assistance. In order to assist them with troubleshooting, you want to copy a couple of the error messages and include them in the e-mail.

Procedures: The procedures below demonstrate how to copy messages out of the status consoles. However, they do not recreate the failure message mentioned in the scenario above.

☐ Open the application status console and the system status console, if necessary.

☐ Start the engine for all solution environments, if necessary.

The system sends a multitude of messages to the application status console. The display looks similar to the following.

☐ Right-click anywhere on the title bar of the application status console window.

☐ Choose the Edit | Mark menu.

The system places a flashing yellow block cursor over the first character of the first row in the window.
☐ Click and hold (do not release) to mark the **beginning of desired block** of text.

The desired block of text does not need to include the first character of the first row.

☐ Drag and release to mark the **end of the desired block** of text.

The system highlights the desired block of text in yellow.

☐ Right-click anywhere on the **title bar** of the **Application Status Console** window.

☐ Choose the **Edit | Copy** menu.

The system copies the marked text to the clipboard, and then returns the text in the window back to its default colors.

☐ Create a **new text document** on the desktop.

☐ Use **Notepad** to open the text document.
In Notepad, choose the **Edit | Paste** menu.

The system copies the clipboard entry into the Notepad document.

```
[09-29 09:42:52.950] [WA] [P_50] [Cleared 0 location user counts.]
[09-29 09:42:52.956] [WA] [P_50] [Cleared 0 log-on flag set endtime.]
[09-29 09:42:52.967] [WA] [P_50] [Cleared 0 assigned work queue entries.]
[09-29 09:42:52.975] [WA] [P_50] [Cleared 38 assigned EMP device entries.]
[09-29 09:42:52.976] [WA] [P_50] [System initialization complete.]
```

Choose the **File | Save** menu.

Choose the **File | Exit** menu.

At this point you could write an e-mail to the HighJump support team describing the nature of the problem, and then attach the Notepad document to it.
Admin Lab Exercise 8:
Log Levels

Introduction

This exercise consists of the following activities:

1. Elevate the Application Log Level with the AWSEM Tool
2. Elevate the Application Log level through the Control INI File
3. Elevate the System Log Level with the SI Manager
4. Elevate the System Log Level through the Control INI File

Background Information

One of the common tasks of an administrator is to determine the root cause of problems. In the previous lab you learned about the logs. The logs are an excellent source of information for identifying problems in the HighJump system. They should be one of the first places you look when you are unable to establish the root cause of a symptom.

Every log message has a severity level associated with it. These severities range from 1 to 5. You can see the severity level in the Detailed Log Report in Advantage Commander under the Log Level column. The five severities are briefly described in the following chart.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Error</td>
</tr>
<tr>
<td>2</td>
<td>Warning</td>
</tr>
<tr>
<td>3</td>
<td>General Information</td>
</tr>
<tr>
<td>4</td>
<td>Diagnostic Information</td>
</tr>
<tr>
<td>5</td>
<td>Trace information and communication information</td>
</tr>
</tbody>
</table>

The log level of the HighJump system determines which log messages are written to the log. When the HighJump system is running at application log level 3 (the default), any application log message with a severity of 1, 2, or 3 is written to the log. When the HighJump system is running at application log level 4, any application log message with a severity of 1, 2, 3, or 4 is written to the log. Application log level 5 operates in a similar manner. You can control the log level for system log messages and application log messages independent of one another.
By default, the HighJump system runs at application log level 3 and system log level 3. However, there may be times when this log level does not provide enough detailed information to troubleshoot a problem. In these cases, you may want to see some of the log messages with a severity level of 4 or 5. The messages with these severity levels will provide many additional details not provided by the others. You can instruct the system to write out log messages with severity level 4 or 5 by changing the application log level or the system log level.

The HighJump system provides four methods for changing the application log level or the system log level. Depending on the situation, some methods may work better than others. The advantages and disadvantages of each method are discussed in the individual activities. The three methods are briefly described in the following chart.

<table>
<thead>
<tr>
<th>Method</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWESM Tool</td>
<td>Advantage Workflow Engine Service Manager</td>
</tr>
<tr>
<td>System Interface Manager</td>
<td>Command-line based tool used for several administrative tasks</td>
</tr>
<tr>
<td>Advantage Commander</td>
<td>Web-based tool used for several administrative tasks (not demonstrated in this manual).</td>
</tr>
<tr>
<td>CONTROL.INI</td>
<td>Control file read by the engine on solution environment startup.</td>
</tr>
</tbody>
</table>

Lastly, elevating a log level can be a useful method of debugging a problem in the HighJump system. However, keep in mind that writing additional log messages consumes additional resources on both the application server and the database server. In some cases, the terminals may experience a performance degradation, when the log level is elevated. This is especially true when you elevate the log level for an entire solution environment or for all devices. As a result, you should not leave the system in an elevated log level any longer than necessary. It is essential to return the system back to log level 3 after you have captured the desired data.

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the Log Levels menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Elevate the Application Log Level with the AWESM Tool

**Background:** The AWESM tool is a multi-function program. In previous exercises, you have seen that it gives you the ability to start and stop solution environments. It also gives you the ability to change the application log level for any device. If you want the log level change to impact exactly one device, then this is the preferred method.

The information captured in application log level 5 can also be derived by using the Visual Debugger tool, which is discussed in a later exercise.

**Situation:** Earlier this morning you promoted a new application into the production environment that contained some changes to the picking business process. Shortly after the promotion, one of the warehouse supervisors calls you on the phone. He explains that the RF readers are exhibiting some abnormal behavior in the picking process. After the user scans the item number, the system correctly displays the container number prompt. And at the top of the container number dialog, the system displays an item description. However, sometimes the item description does not reflect the description of the item number from the previous scan. The problem is sporadic in nature.

You are certain that it is an application problem, but you do not have the Advantage Architect skills to determine the root cause. You call on the application development team to solve the problem. They are not able to reproduce the problem in a testing environment, and they have requested some elevated log information from you in order to continue the troubleshooting process. Because the problem is sporadic, you choose to elevate the application log level for a period of time for a single RF reader in hopes that the reader will encounter the problem during that timeframe.

**Current State:** Whenever you make a change to the HighJump system it is good practice to validate that the change yielded the desired outcome. However, if you don’t know the initial state of the system, it is difficult to determine if the change gave the desired result. The next couple steps familiarize you with the current state of the system so that you have a basis of comparison when you need to determine if the change was successful.

- Open the **Application Status Console**, if necessary.
- Open the **HighJump One Platform UI**, if necessary.
- Click the **Menu | Advantage Commander | Advantage Commander | Log | Summary Log Report** menu.
- Click the **Submit** button in the **Action Bar**.

The system displays the Log Messages Summary View page.

- Review the current contents of the page.
- Click on the **Last Page** button at the bottom of the page.
Note the current number of pages returned by the search.

Procedures: The procedures below demonstrate how to elevate the application log level for one reader using the AWESM tool. However, they do not reproduce the scenario of the invalid item description in the picking process.

- Open the Advantage Workflow Engine Service Manager, if necessary.
- Click the WA solution environment, in order to highlight it.
- Click the Sessions button.

The system opens another window which displays all of the sessions for the WA solution environment. This window shows terminals (RADIO_01), and it shows other "system-type" sessions. These "system-type" sessions are essentially sessions running in the background. (If the system does not show any sessions, then choose the Show All option in the View drop-down list.)

- Click the RADIO_01 device, in order to highlight it.
☐ Click the **Log Levels** button.

The system displays the current log levels for the device.

![](image)

☐ Choose 5 – **Trace** in the **Application** drop-down list.

☐ Click the **OK** button.

---

**Test Cases:** The final step in this process is to validate that the changes work as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Login to Virtual Terminal <strong>RADIO_01</strong> and execute a couple iterations of</td>
<td>The system floods the Application Status Console with detailed log messages.</td>
</tr>
<tr>
<td></td>
<td>the <strong>Location Status</strong> business process. Browse the **Application Status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Console.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Browse the <strong>Summary Log Report</strong> in Advantage Commander.</td>
<td>The Log Message Summary View page contains a significant increase in the number of log messages. Also the nature of the messages is very detailed.</td>
</tr>
</tbody>
</table>
Follow Up: The log data will look similar to the following screenshot. Note the large number of pages in the view. Also note that every time a variable is updated in memory the system writes a log message. Additionally, as you scroll through the messages you will see the database queries as well as the line numbers and names of the various process objects.

At this point you could export the log through Advantage Commander or the Log Export Utility, and then send the log to the application development team. This is a significant amount of information and will be invaluable to the application development team as they continue to troubleshoot the issue.

Elevating the log level, and writing additional messages, consumes resources on both the application and the database servers. The solution can experience a performance degradation when the log level is elevated over an extended time. Therefore, it is essential to return the system back to log level 3 after you have captured the desired data. Complete the steps that follow to avoid issues with your learning environment.

You have captured the desired information in the log. There is no longer a need to keep RADIO_01 in an elevated log status.

☐ Use the AWESM tool to set the Application Log Level to 3 for RADIO_01.
Activity 2: Elevate the Application Log Level through the Control INI File

**Background:** The control.ini file is a text file that holds a variety of parameters. Among other parameters, the file includes the default application log level for a solution environment. The system reads this parameter when a given solution environment starts.

The AWESM tool is capable of elevating the application log level for a specific device and other managers. However, if the problem occurs before you are able to see the device in the AWESM tool, then you may need to elevate the log level using another method. The methods described in this activity provide an option for elevating the application log level when the AWESM tool does not suffice. (This may be the case when the problem occurs as part of engine startup.)

As with any elevated log level, the HighJump system will consume system resources on both the application server and the database server. You can expect a general performance degradation when you raise the log level. This is especially true with the control.ini method, as it impacts all defined devices (RF terminals, Virtual Terminals...) as well as other managers for a given solution environment. This option is not recommended in a production environment.

Lastly, if you use this method, you must shutdown the solution environment and then restart the solution environment before the elevated log level will take effect. This is another reason to exercise caution when operating in production.

**Situation:** You have a background process (developed in Advantage Architect) that prints reports. This background process initiates when engine starts. You manually place a couple report requests into the queue and then start the engine in the test environment. The system prints all of the reports in the queue except for the first one. After some research you learn that the background process ran into some problems immediately after the engine started, and then it corrected itself. You want to elevate the log level in order to assist with the troubleshooting process. However, by the time you could use the AWESM tool it would be too late. Additionally you are working in the test environment. You opt to raise the log level using control.ini.

**Current State:** Whenever you make a change to the HighJump system it is good practice to validate that the change yielded the desired outcome. However, if you don't know the initial state of the system, it is difficult to determine if the change gave the desired result. The next couple steps familiarize you with the current state of the system so that you have a basis of comparison when you need to determine if the change was successful.

1. Open the **HighJump One Platform UI**, if necessary.
2. Click the **Menu | Advantage Commander | Advantage Commander | Log | Summary Log Report** menu.
3. Click the **Submit** button in the **Action Bar**.

The system displays the Log Messages Summary View page.
Review the current contents of the page.

Click on the Last Page button at the bottom of the page.

Note the current number of pages returned by the search.

Procedures: The procedures below demonstrate how to elevate the application log level at a solution environment level using the control.ini file. However, they do not reproduce the scenario of the broken replenishment page.

Shutdown the engine for the WAPROCESSORS solution environment.

Open Windows Explorer.

Navigate to the D:\ProgramData\HighJump Software\CONTROL\CONTROL1 folder.

Double click the CONTROL.INI file.

Scroll down to the block with the title SOLUTION_WAPROCESSORS

```
[SOLUTION_WAPROCESSORS]
ApplicationSource =
ApplicationLogLevel = 3
DefaultSQLServerName = HJU-VLAB
DefaultSQLServerDBName = AAD
DefaultSQLServerUserID = HJS
DefaultSQLServerPassword = HJSPASS#1
```
In the SOLUTION_WAPROCESSORS block change the ApplicationLogLevel value to 5.

```
[SOLUTION_WAPROCESSORS]
ApplicationLogLevel = 5
DefaultSQLServerName = HJU-VLAB
DefaultSQLServerDBName = AAD
DefaultSQLServerUserID = HJS
DefaultSQLServerPassword = HJSPASS#1
```

Choose the File | Save menu.

Choose the File | Exit menu.

Start the engine for the WAPROCESSORS solution environment.

**Test Cases:** The final step in the process is to validate that the change worked as expected. Since the CONTROL.INI change also impacts the entire solution environment, you do not need to execute a replenishment in order to validate that the system is running in application log level 5. The WA solution environment contains a handful of background processes that initiate when the engine starts. If the change worked correctly, these processes should already be generating a multitude of additional log messages.

The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wait for one minute. Then browse the Application Status Console.</td>
<td>The system floods the Application Status Console with detailed log messages every 5 seconds.</td>
</tr>
<tr>
<td></td>
<td>Browse the Summary Log Report in Advantage Commander.</td>
<td>The Log Message Summary View page contains a significant increase in the number of log messages. Also the nature of the messages is very detailed.</td>
</tr>
</tbody>
</table>
Follow Up: The log data will look similar to the following screenshot. Note that the number of pages in the view has once again significantly increased from the previous viewing. Also note that the content of the messages is similar to the log level 5 content from the previous activity.

At this point you could export the log through Advantage Commander or the Log Export Utility, and then send the log to the application development team. This is a significant amount of information and will be invaluable to the application development team as they continue to troubleshoot the issue.

Elevating the log level, and writing additional messages, consumes resources on both the application and the database servers. The solution can experience a performance degradation when the log level is elevated over an extended time. Therefore, it is essential to return the system back to log level 3 after you have captured the desired data. Complete the steps that follow to avoid issues with your learning environment.

You have captured the desired information in the log. There is no longer a need to keep the entire WA Processors solution environment in an elevated log status.

- Shutdown the engine for the WAPROCESSORS solution environment.
- Change the ApplicationLogLevel value in the CONTROL.INI file to 3 for the WAPROCESSORS solution environment.
- Start the engine for the WAPROCESSORS solution environment.
Activity 3: Elevate the System Log Level with the AWESM Tool

**Background:** The AWESM tool is a multi-function program. In previous exercises, you have seen that it gives you the ability to start and stop solution environments. It also gives you the ability to change the system log level for any device. Generally speaking, this is the preferred method for changing the system log level.

**Situation:** One of the warehouse supervisors calls you on the phone. He indicates that approximately half of the RF readers are experiencing some odd behavior. Approximately every two hours, this subset of readers times out for one minute. Then the readers reconnect and the readers return to the exact state they were in prior to the timeout.

You look at the log and see some pertinent log entries, but you are not able to interpret them. After a bit more research you determine that all of the RF readers with the timeout are attached to the same socket. The other readers that work correctly are attached to a separate socket. After calling the HighJump Support team, they request an elevated system log for the socket in question.

**Current State:** Whenever you make a change to the HighJump system it is good practice to validate that the change yielded the desired outcome. However, if you don’t know the initial state of the system, it is difficult to determine if the change gave the desired result. The next couple steps familiarize you with the current state of the system so that you have a basis of comparison when you need to determine if the change was successful.

- Open the HighJump One Platform UI, if necessary.
- Click the Menu | Advantage Commander | Advantage Commander | Log | Summary Log Report menu.
- Click the Submit button in the Action Bar.

The system displays the Log Messages Summary View page.

- Review the current contents of the page.
- Click on the Last Page button at the bottom of the page.
- Note the current number of pages returned by the search.
Procedures: The procedures below demonstrate how to elevate the system log level for one device using the AWESM tool. However, they do not reproduce the scenario of the reader timeout.

☐ Open the Advantage Workflow Engine Service Manager, if necessary.

☐ Start the engine for the WA solution environment, if necessary.

☐ Click the WA solution environment, in order to highlight it.

☐ Click the Sessions button.

The system opens another window which displays all of the sessions for the WA solution environment. This window shows terminals (RADIO_01), and it shows other "system-type" sessions. These "system-type" sessions are essentially sessions running in the background. (If the system does not show any sessions, then choose the Show All option in the View drop-down list.)

☐ Click the RADIO_01 device, in order to highlight it.

☐ Click the Log Levels button.

The system displays the current log levels for the device.
Choose 5 – Trace in the System drop-down list of the Process Manager section.

Click the OK button.

Test Cases: The final step in this process is to validate that the changes work as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Login to Virtual Terminal RADIO_01 and execute a couple iterations of the Location Status business process.</td>
<td>System exhibits normal behavior on the Virtual Terminal.</td>
</tr>
<tr>
<td></td>
<td>Browse the System Status Console</td>
<td>The System Status Console remains relatively quiet.</td>
</tr>
<tr>
<td></td>
<td>Browse the Summary Log Report in Advantage Commander.</td>
<td>The Log Message Summary View page contains a significant increase in the number of log messages. Also the nature of the messages is very different from the application messages.</td>
</tr>
</tbody>
</table>

The log data will look similar to the following screenshot. Note that the content of the messages is noticeably different from the application log level 5 messages. You probably did not notice any performance degradation with this activity. Nor did the number of log message increase significantly. This is because only a handful of messages are logged for each communication packet. If there were
multiple terminals on the socket all running processes, you likely would have experienced some performance degradation.

At this point you could export the log through Advantage Commander or the Log Export Utility, and then send the log to the HighJump Support team. This is a significant amount of information and will be invaluable to the support team as they continue to troubleshoot the issue.

Elevating the log level, and writing additional messages, consumes resources on both the application and the database servers. The solution can experience a performance degradation when the log level is elevated over an extended time. Therefore, it is essential to return the system back to log level 3 after you have captured the desired data. *Complete the follow steps to avoid issues with your learning environment.*

You have captured the desired information in the log. There is no longer a need to keep the socket in an elevated log status.

☐ Use the **AWESM tool** to set the **System Log Level** to 3 for **RADIO_01**.
Activity 4: Elevate the System Log Level through the Control INI File

**Background:** The control.ini file is a text file that holds a variety of parameters. Among other parameters, the file includes the default system log level. The system reads this parameter when the entire set of solution environments start.

As with any elevated log level, the HighJump system will consume system resources on both the application server and the database server. You can expect a general performance degradation when you raise the log level. This is especially true with the control.ini method, as it impacts all devices. Use this option with extreme caution in a production environment.

Lastly, if you use this method, you must shutdown all solution environments and then restart the solution environments before the elevated log level will take effect. This is another reason to exercise caution when operating in production.

**Situation:** You are the HighJump administrator for a large multi-warehouse operation. As the company begins the data setup phase of the project, you have been tasked with defining over two hundred RF readers in Advantage Commander and then testing each one. Your plan is to define them in the test environment first, and then use a script to port the device definitions from the test environment over to the production environment. In order to accommodate the aggressive growth plans, you decide to spread the reader definitions across four different sockets. You define all of the readers in Advantage Commander; you create all of the database logins; and you configure all of the physical hardware.

After you have defined all of the devices, you turn on the engine and test them, and none of them work. All of the readers display the error “Advantage Engine Down.” You look at the log and see some pertinent log entries for the sockets, but you are not able to interpret them. After calling the HighJump Support team, they request an elevated system log for the sockets. Since the problems are in the test environment, you decide to take the easy path and elevate the log through the control.ini file.

**Current State:** Whenever you make a change to the HighJump system it is good practice to validate that the change yielded the desired outcome. However, if you don’t know the initial state of the system, it is difficult to determine if the change gave the desired result. The next couple steps familiarize you with the current state of the system so that you have a basis of comparison when you need to determine if the change was successful.

- Open the HighJump One Platform UI, if necessary.
- Click the **Menu | Advantage Commander | Advantage Commander | Log | Summary Log Report** menu.
- Click the **Submit** button in the **Action Bar**.

The system displays the Log Messages Summary View page.

- Review the current contents of the page.
Click on the Last Page button at the bottom of the page.

Note the current number of pages returned by the search.

**Procedures:** The procedures below demonstrate how to elevate the system log level with the control.ini file. However, they do not reproduce the scenario of the problematic readers.

- Shutdown the engine for all solution environments.
- Open Windows Explorer.
- Navigate to the D:\ProgramData\HighJump Software\CONTROL\CONTROL1 folder.
- Double click the CONTROL.INI file.
- Scroll down to the block with the title CONTROLLER

In the CONTROLLER block change the SystemLogLevel value to 5.
Choose the **File | Save** menu.

Choose the **File | Exit** menu.

Start the **engine** for all solution environments.

**Test Cases:** The final step in the process is to validate that the change worked as expected. The system constantly attempts to communicate with the sockets. If the change worked correctly, the system should already be generating a multitude of additional log messages.

The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Wait for one minute. Then browse the <strong>Summary Log Report</strong> in Advantage Commander.</td>
<td>The Log Message Summary View page contains a significant increase in the number of log messages. Also the nature of the messages is very detailed.</td>
</tr>
</tbody>
</table>

**Follow Up:** The log data will look similar to the following screenshot. Note that the number of pages in the view has once again significantly increased from the previous viewing. Also note that the content of the messages is similar to the log level 5 content from the previous activity.
At this point you could export the log through Advantage Commander or the Log Export Utility, and then send the log to the HighJump support team. This is a significant amount of information and will be invaluable to them as they continue to troubleshoot the issue.

Elevating the log level, and writing additional messages, consumes resources on both the application and the database servers. The solution can experience a performance degradation when the log level is elevated over an extended time. Therefore, it is essential to return the system back to log level 3 after you have captured the desired data. Complete the steps that follow to avoid issues with your learning environment.

You have captured the desired information in the log. There is no longer a need to keep all solution environments in an elevated log status.

- [ ] Shutdown the engine for the all solution environments.
- [ ] Change the SystemLogLevel value in the CONTROL.INI file to 3.
- [ ] Start the engine for all solution environments.
Admin Lab Exercise 9: Managers

Introduction

This exercise consists of the following activities:

1. Restart a manager with the AWESM tool.

Background Information

When you start the engine, the system initiates a series of managers. Some of these managers control communications between the terminals and the application servers. Some of these managers govern the Crystal reports that are sent to printers. Yet other managers regulate the data passed between the application server and a host system.

Occasionally, these managers may encounter a problem from which they cannot recover. When this happens somebody must intervene and take corrective action. As an administrator, you have a couple options. You could cycle the engine. While this method often restores the manager, it also requires all material handlers to cease work for several minutes. Because of this severe limitation, this method is not recommended unless absolutely necessary.

Another option for restoring the manager is to stop and start just the problematic manager. This method is often successful, and it is far superior to cycling the entire engine. The activities below demonstrate how to cycle an individual manager.

While there are a couple different methods for cycling an individual manager, the activities below demonstrate how to do this through the AWESM tool. All HighJump managers can be started and stopped through this tool.

Stopping a manager is a risky activity. If you do not do it correctly, or if you stop the wrong manager, you may end up with disastrous results. This is especially true of process managers and device managers, which pass data between the terminals and the application server. HighJump recommends stopping a manager only under the direction of the HighJump support team.

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the Restarting Manager menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Restart a Manager with the AWESM Tool

Scenario: Your shipping process generates a significant number of Crystal reports that print on a laser printer in the shipping office. The warehouse manager calls you on phone to report that the shipping reports are no longer printing. He asks you to find a solution as soon as possible, as the shipping reports are critical to the operation. You review the logs and you see several errors related to the event manager. You contact the HighJump support team, and they agree that the problem is most likely related to the event manager. They ask you to stop and then start the event manager.

Procedures: The procedures below demonstrate how to start and stop the event manager with the AWESM tool. However, they do not recreate the problem with the Crystal reports.

☐ Start the engine, if necessary.

It is not necessary to open the System Status Console in order to stop and start a manager. However, this activity includes this step in order to give you a visual indication that the state of the manager has changed.

☐ Open the System Status Console.

☐ Open the Advantage Workflow Engine Service Manager, if necessary.

☐ Click the WA solution environment, in order to highlight it.

☐ Click the Sessions button.

The system opens another window which displays all of the sessions for the WA solution environment. This window shows terminals (RADIO_01), and it shows other "system-type" sessions. These "system-type" sessions are essentially sessions running in the background. (If the system does not show any sessions, then choose the Show All option in the View drop-down list.)

☐ Click the Event Manager device, in order to highlight it.

☐ Click the Stop Event Mgr button.

The system displays a dialog asking if you really want to shut down the event manager.
☐ Click the Yes button.

The System Status Console that indicates that the event manager was shut down successfully. The [EVENTMGR] element in the display refers to the event manager.

Starting the event manager is executed in a very similar manner.

☐ Click the Event Manager device, in order to highlight it.

☐ Click the Start Event Mgr button.

☐ Click the X in the upper right corner of the Sessions window, in order to close it.

The System Status Console indicates that the event manager was started successfully. The system starts each of the threads associated with the event manager. These threads are indicated the by E_01 and the E_02 notation.

At this point the event manager has been stopped and started. The warehouse manager can run a test to determine if this process solved the problem with the Crystal reports.
Admin Lab Exercise 10: Advantage Link

Introduction

This exercise consists of the following activities:

1. Import the Text File
2. Research the Advantage Link Import Results
3. Modify the Advantage Link Import Data
4. Resubmit an Advantage Link Import Batch
5. Research the Advantage Link Import Resubmission
6. Create an Inventory Adjustment Export Record
7. Research the Advantage Link Export Results
8. Modify the Advantage Link Export Data
9. Resubmit an Advantage Link Export Batch
10. Research the Advantage Link Export Resubmission
Background Information

Advantage Link is a software layer that sits between Warehouse Advantage and the host system. It allows Warehouse Advantage to communicate with a variety of host systems. Advantage Link can communicate with a host system via a text file, XML, standard SAP iDocs, as well as a handful of other methods. The diagram below shows the data flow for exports and imports.

Warehouse Advantage Exports

Warehouse Advantage Imports

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the ADV Link menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Import the Text File

**Situation:** Your Advantage Link software layer communicates with a host system via a text file. The host system has recently sent two text files containing item master information to the Advantage Link layer.

**Procedures:** The training environment does not contain a host application. In this activity, rather than using a host system, you will manually emulate a host system that sends two item master text files to Advantage Link.

The training folder on the desktop contains two item master files which were manually created by the instructor. These text files are identical to ones created by a host system. Before you do anything with the files, you will first review the contents of each one.

☐ Open the **Training** folder on the desktop.

☐ Navigate to the **AdvLink \ Items** folder.

The folder contains two text files: SCAITEM_BATCH_1.DAT and SCAITEM_BATCH_2.DAT. These are the two files you will review and eventually import into the system.

☐ Open the **Notepad** application.

☐ Choose the **File | Open** menu option.

☐ Navigate to the **Desktop \ Training \ AdvLink \ Items** folder.

☐ Choose **All Files** in the drop-down list in the lower-right corner.

☐ Click **SCAITEM_BATCH_1.DAT** in order to highlight the filename.

☐ Click the **Open** button.

The system displays the contents of the text file.
Each row in the file contains the data for one item master record. The actual item numbers begin in position 3 and end in position 10. The item descriptions appear in the right side of the screenshot. As you scroll from left to right you will see the other item master attributes in the file.

In order to demonstrate some features included in the HighJump One Platform UI, some of the item records in the file contain dirty data that will be trapped by the item import process. The chart below shows the item number, the relevant attributes, and the corresponding problem for each record.

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Item Description</th>
<th>Warehouse ID</th>
<th>Class ID</th>
<th>Lot Control Flag</th>
<th>Problem Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYITEM01</td>
<td>SCREWDRIVER</td>
<td>77</td>
<td>--</td>
<td>N</td>
<td>Invalid Warehouse</td>
</tr>
<tr>
<td>MYITEM02</td>
<td>HAMMER</td>
<td>01</td>
<td>FG</td>
<td>N</td>
<td>--</td>
</tr>
<tr>
<td>MYITEM03</td>
<td>CHAINSAW</td>
<td>01</td>
<td>XY</td>
<td>N</td>
<td>Invalid Class</td>
</tr>
<tr>
<td>MYITEM04</td>
<td>CROWBAR</td>
<td>01</td>
<td>FG</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>MYITEM05</td>
<td>WRENCH</td>
<td>01</td>
<td>FG</td>
<td>W</td>
<td>Invalid Lot Control Flag</td>
</tr>
</tbody>
</table>

In summary, there are five item master records in the first file. Three of the records contain dirty data, and two of the records contain clean data.

Now you will review the contents of the second file.

☐ In Notepad, choose the File | Open menu option.

☐ Choose All Files in the drop-down list in the lower-right corner.

☐ Click SCAITEM_BATCH_2.DAT in order to highlight the filename.

☐ Click the Open button.

The system displays the contents of the text file.
Like the first file, each row contains the data for one item master record. The format of the data is similar as well. However, in this file none of the data has any issues. All of the records are clean and will import successfully. The chart below shows the item number and the relevant attributes for each record.

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Item Description</th>
<th>Warehouse ID</th>
<th>Class ID</th>
<th>Lot Control Flag</th>
<th>Problem Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYITEM06</td>
<td>PLUNGER</td>
<td>01</td>
<td>FG</td>
<td>N</td>
<td>--</td>
</tr>
<tr>
<td>MYITEM07</td>
<td>AWL</td>
<td>01</td>
<td>FG</td>
<td>N</td>
<td>--</td>
</tr>
<tr>
<td>MYITEM08</td>
<td>PAINTBRUSH</td>
<td>01</td>
<td>FG</td>
<td>N</td>
<td>--</td>
</tr>
</tbody>
</table>

In summary, there are three item master records in the second file, and all of the records are clean.

You have seen that the raw data in the item import files is for items MYITEM01 through MYITEM08. Eventually, you will import these items and then validate that the system imported them successfully. However before you launch the import you will verify that none of these items currently reside in the Warehouse Advantage item master.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Supply Chain Advantage | Warehouse Advantage | Items menu.

The system opens the Search Items page that will return item master data.
Admin Lab Exercise 10:
Advantage Link

All of the items included in the item import files start with the text “MYITEM”. Those are the only items in which we are interested, so you include that in the search criteria.

☐ Enter MYITEM in the Item Number edit box.

☐ Click the Query button in the Action Bar.

At this point in the process there aren’t any items which have item numbers beginning with MYITEM, and the system indicates that on the page.

The text files containing the item import data are currently located in a folder created by the instructor. Now that you have reviewed the data in the text files and the data in the item master table, you will copy the text files into a different folder, and the HighJump system will import them. In a production system, the host would automatically place the files into the folder. The training environments do not include a host, so you will you place the files into the folder manually.

☐ In Notepad, choose the File | Exit menu option.

☐ In the previous instance of Windows Explorer, copy SCAITEM_BATCH_1.DAT and SCAITEM_BATCH_2.DAT
Navigate to the D: \ ProgramData \ HighJump Software \ IMPORTS \ ITEM folder.

Paste SCAITEM_BATCH_1.DAT and SCAITEM_BATCH_2.DAT into this folder.

The folder should look like the following screenshot.

In order to import the data, the solution environments in the AWESM tool need to be running (specifically ADVLINKFLATFILE, WA, and WAPROCESSORS). The status consoles are not required for a successful import, but they will provide some valuable information regarding the import.

Start all solution environments in the AWESM tool, if necessary.

Open the Application Status Console, if necessary.

Even though the solution environments are running, the HighJump system does not import the files, because it is looking for a very specific filename. With the base configuration, HighJump is expecting the host to create a file named SCAITEM.DAT.

In Windows Explorer, rename SCAITEM_BATCH_1.DAT to SCAITEM.DAT.

Wait a couple seconds.

In the base configuration, the HighJump system polls for SCAITEM.DAT approximately once a minute. When it finds the file, the system imports the data, writes some information to the platform log and the status consoles, then renames the file.
The system writes several messages to the application status console (and the platform log) regarding the import. The console indicates when the import started and when it finished. The rows in yellow indicate that the import process detected some problems with the item master data. The application status console should look similar to the following screenshot.

[Application Status Console screenshot]

☐ In the Application Status Console, scroll to the right.

The system will display a complete description of the problems it detected with the item master data. The descriptions in the console correspond with the review of the raw data you performed earlier in this activity. The application status console should look similar to the following screenshot.

[Application Status Console screenshot]

After the system imported the data, the system renamed the import file and gave it a name with a sequentially numbered extension. The files in the folder should look like the following screenshot.

[File folder screenshot]

The HighJump system processed the first import file. Next you will walk through similar steps with the second text file.

☐ In Windows Explorer, rename SCAITEM_BATCH_2.DAT to SCAITEM.DAT.

☐ Wait a couple seconds.
Once again, the system imports the data in the file, writes some information to the platform log and the status consoles, and then renames the file. The application status console now contains information regarding the second import. It should look similar to the following graphic.

Note the absence of yellow error records in the application status console for the second import. The second file did not contain any dirty data. Therefore, the system did not write any error messages to the console. This corresponds with the review of the raw data you performed earlier in this activity.

After the system imported the second data file, the system renamed all existing backup files by incrementing the extension by 1. Then it renamed the newly imported file as SCAITEM.001. The files in the folder should look like the following screenshot.

The text in the application status console indicates that some of the imports were successful and others were not, but it does not indicate which items fall into each category. Because some of the imports were successful, you will be able to see them in the Warehouse Advantage item master web pages.

- Open the **HighJump One Platform UI**, if necessary.
- Click the **Menu | Supply Chain Advantage | Warehouse Advantage | Items** menu.
The system opens the Search Items page that will return item master data.

![Search Items](image)

All of the items included in the item import files start with the text “MYITEM”. Those are the only items in which we are interested, so you include that in the search criteria.

- Enter **MYITEM** in the **Item Number** edit box.
- Click the **Query** button in the **Action Bar**.

The system displays five item master records that were imported successfully.

![Items](image)

In this activity you imported two item master text files into the HighJump system. You saw that some were imported successful and others failed. However, you didn’t see any explanation for the failed records. In the next activity, you will learn how to use the Advantage Link Administration web pages to gather detailed information about why some of the records failed.
Activity 2: Research the Advantage Link Import Results

Situation: One of the receivers on the receiving dock has indicated that he cannot receive a certain item into the warehouse because the RF terminals display an error that reads “INVALID ITEM”. The receiver looked at the item master table through the HighJump One Platform UI and did not find the given item. The host administrator tells you that the host sent that new item number to the HighJump system in a recent import file. It is your job to understand what happened to that item number during the import.

Procedures: In the previous activity you imported two text files into the HighJump system. And you reviewed the import-related information in the application status console. The information in the console is useful, but it has a limited buffer. At some point those import-related messages will no longer be visible in the console. In this activity, you will view similar information in the HighJump One Platform UI regarding the item imports. Here you will see that the Advantage Link Administration pages provides more detailed information than the application status console.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Supply Chain Advantage | Advantage Link Administration | Imports | Item menu.

The system displays a web page which allows you to enter specific search criteria. The goal is to find the two imports that you imported in the previous activity. You can search for the imported data by entering a partial item number, a date range, or several other criteria. In this case the sample size is relatively small and there is no need to enter any search criteria.

☐ Click the Query button in the Action Bar.

The system displays a list of all imports that meet the search criteria.
The chart below describes each one of the columns on the Item Import page.

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Group ID</td>
<td>A system-assigned identifier. There is one Host Group ID per text file.</td>
</tr>
<tr>
<td>Record Create Date</td>
<td>The date the system imported the items.</td>
</tr>
<tr>
<td>Master</td>
<td>The number of item master records included in the import.</td>
</tr>
<tr>
<td>UOM</td>
<td>The number of unit of measure records included in the import.</td>
</tr>
<tr>
<td>Resubmit</td>
<td>Allows you to reimport the data. (More on this later.)</td>
</tr>
</tbody>
</table>

Note that there are two rows in the Item Import page, and each row corresponds with one of the text files that was imported. One text file contained five item master records and the other text file contained three item master records.

In addition to viewing the Host Group ID and the import date, you can also see the item master data that was imported from the host. However, you first must navigate to a different Advantage Link Administration page.

☐ Click the number 5 under the **Master** column.

The system displays item master import data provided by the host system in the first import file.
The chart below describes several of the columns on the Item Master Import page.

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Item Master ID</td>
<td>A system-assigned number. There is one Host Item Master ID per item.</td>
</tr>
<tr>
<td>Record Create Date</td>
<td>The date the system imported the item.</td>
</tr>
<tr>
<td>Processing Code</td>
<td>New, Update, or Delete – Indicates whether the HighJump system should insert, update, or delete the item master record. (The value is provided by the host system.)</td>
</tr>
<tr>
<td>Import Status</td>
<td>Error or Success – Indicates whether the HighJump system successfully imported the record.</td>
</tr>
<tr>
<td>Error Message</td>
<td>If the import of the item failed, the reason for the failure appears here.</td>
</tr>
<tr>
<td>Warehouse ID, Client, Code, Item Number (and following)</td>
<td>Item master data provided by the host system.</td>
</tr>
</tbody>
</table>

Note that MYITEM01, MYITEM03, and MYITEM05 failed to import. The reason for the failures appear under the Error Message column, and these messages correspond with the research you did in the previous activity. The graphic below shows the relationship between the errors in the application status console from the previous activity and the errors in Advantage Link Administration.
Use the horizontal scroll bar to view the remainder of the data sent by the host.

You have researched the data which the host sent in the first import file. Now you will research the data in the second import file.

Click the Back to Item Import button in the Action Bar.

The system displays the Item Import window again.

Click the number 3 under the Master column.

The system displays item master import data provided by the host system in the second import file.
Note that all of the records included in the second file imported successfully. This corresponds with the research you did in the previous activity.

☐ Use the horizontal scroll bar to view the remainder of the data sent by the host.
Activity 3: Modify the Advantage Link Import Data

**Situation:** You have researched the Advantage Link import data and you have discovered that some of the items did not import successfully because the host sent dirty data. Due to some critical events on the warehouse floor you need HighJump to import those items as soon as possible. You could work with the host administrator to fix and resend the text file. But you decide that option would take too much time. Instead, you opt to fix the data on the HighJump side first, and then inform the host administrator about the situation.

**Procedures:** In the previous activity you used the HighJump One Platform UI to view the item import errors associated with the first import file. You will now use the Advantage Link Administration pages to correct the dirty data from the item import, with the intent that the data will be resubmitted (or reimported) in the future.

1. Open the HighJump One Platform UI, if necessary.
2. Click the Menu | Supply Chain Advantage | Advantage Link Administration | Imports | Item menu.
3. Click the Query button in the Action Bar.
4. Click the number 5 under the Master column.

The system displays the 5 item master import records provided by the host system in the first import file.

<table>
<thead>
<tr>
<th>#</th>
<th>Host Item Master ID</th>
<th>Record Create Date</th>
<th>Processing Code</th>
<th>Import Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10/6/2016</td>
<td>New</td>
<td>Error</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>10/6/2016</td>
<td>New</td>
<td>Success</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>10/6/2016</td>
<td>New</td>
<td>Error</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>10/6/2016</td>
<td>New</td>
<td>Success</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>10/6/2016</td>
<td>New</td>
<td>Error</td>
</tr>
</tbody>
</table>

The first row contains the import data associated with MYITEM01. However, the import of this item failed because of an invalid warehouse. The system indicates in the Warehouse ID column that the host provided a warehouse value of 77.
Since this value does not reside in the warehouse master table, it did not import. In order to correct the data you will change the warehouse value from 77 to 01.

- Click the number 1 in the first column from the left.

The system displays all of the item import attributes, and it gives you the opportunity to change them.

- Enter 01 in the Warehouse ID edit box.
- Click the Update button in the Action Bar.

The system updates the data and returns to the Item Master Import window which displays the new value for the Warehouse ID.
The third row contains the import data associated with MYITEM03. However, the import of this item failed because of an invalid class ID. The system indicates in the Class ID column that the host provided a class ID value of XY.

<table>
<thead>
<tr>
<th>Reorder Qty</th>
<th>Class ID</th>
<th>Pick Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td>1121</td>
</tr>
<tr>
<td>0</td>
<td>FG</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>XY</td>
<td></td>
</tr>
</tbody>
</table>

Since this value does not reside in the class master table, it did not import. In order to correct the data you will change the class value from XY to FG (finished goods).

- Click the number 3 in the first column from the left.

The system displays all of the item import attributes, and it gives you the opportunity to change them.

- Enter FG in the Class ID edit box.
- Click the Update button in the Action Bar.

The system updates the data and returns to the Item Master Import window which displays the new value for the Class ID.
The fifth row contains the import data associated with MYITEM05. However, the import of this item failed because of an invalid lot control flag. The system indicates in the Lot Control column that the host provided a warehouse value of W.

Since this value is not a valid flag, it did not import. In order to correct the data you will change the lot control value from W to N (not lot controlled).

☐ Click the number 5 in the first column from the left.

The system displays all of the item import attributes, and it gives you the opportunity to change them.

☐ Enter N in the Lot Control edit box.
☐ Click the **Update** button in the **Action Bar**.

The system updates the data and returns to the Item Master Import window which displays the new value for the Lot Control.

<table>
<thead>
<tr>
<th>Shelf Life</th>
<th>Lot Control</th>
<th>Reorder Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N</td>
<td>50</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td><strong>0</strong></td>
<td><strong>N</strong></td>
<td><strong>0</strong></td>
</tr>
</tbody>
</table>
Activity 4: Resubmit an Advantage Link Import Batch

Situation: The host system sent HighJump an item import file that contained dirty data. You identified the dirty data and you corrected it in the Advantage Link Administration pages in the HighJump One Platform UI. Now you want the HighJump system to reimport the data.

Procedures: In the previous activity you corrected some dirty item import data through the Advantage Link Administration pages. In this activity you will reimport the modified records into the HighJump system. The Advantage Link Administration pages use the term “resubmit”. First you will validate that the failed items do not exist in the item master table. Then you will resubmit the Advantage Link batch. Then you will revisit the item master table to validate that the import was finally successful.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Supply Chain Advantage | Warehouse Advantage | Items menu.

The system opens the Search Items page which eventually leads to the item master data.

All of the items included in the item import files started with the text “MYITEM”. Those are the only items in which we are interested, so you include that in the search criteria.

☐ Enter MYITEM in the Item Number edit box.

☐ Click the Query button in the Action Bar.

The system displays all items in the item master table that begin with MYITEM. There are 5 of them. Note that MYITEM01, MYITEM03, and MYITEM05 are not among them.
Now that you have reviewed the item master data, you will resubmit that items which you corrected in the previous activity.

☐ Click the Menu | Supply Chain Advantage | Advantage Link Administration | Imports | Item menu.

☐ Click the Query button in the Action Bar.

The system displays the Item Import page with the two batches (groups) of items that were imported. Note the RESUBMIT link on the far right side. If you use this link, the system will resubmit the entire batch (group). That is, the system will attempt to reimport all of the items in the batch. That would include any dirty items that you corrected as well as all of the clean records in that batch.

In some cases resubmitting an entire batch (group) is desirable. In this example, you will get more granular. You will only resubmit the items that you corrected.

☐ Click the number 5 under the Master column.

The system displays item master import data provided by the host system in the first import file.
You corrected MYITEM01, MYITEM03, and MYITEM05 in a previous activity. Those are the only items that you want to resubmit for import.

☐ Press and hold the Ctrl on the keyboard.

☐ Use the mouse to multi-select the MYITEM01, MYITEM03, and MYITEM05 rows.

The page should look like the following screenshot with rows 1, 3, and 5 highlighted in green.

☐ Click the Resubmit button in the Action Bar.
The system briefly displays a “Data was successfully resubmitted” bubble near the lower right corner of the screen.

You corrected the dirty data. Then you resubmitted the items. Now you will verify that the items were successfully imported into the item master table.

☐ Click the Menu | Supply Chain Advantage | Warehouse Advantage | Items menu.

The system opens the Search Items page that will return item master data.

All of the items included in the item import files start with the text “MYITEM”. Those are the only items in which we are interested, so you include that in the search criteria.

☐ Enter MYITEM in the Item Number edit box.

☐ Click the Query button in the Action Bar.

The system displays all items in the item master table that begin with MYITEM. Note that MYITEM01, MYITEM03, and MYITEM05 are now included in the item master list. On the first pass, the system successfully imported 2 items from the first file and 3 items from the second file. Then you corrected the data and resubmitted 3 items. That makes a total of eight.
<table>
<thead>
<tr>
<th>#</th>
<th>Warehouse ID</th>
<th>Item Number</th>
<th>Inventory Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01</td>
<td>MYITEM01</td>
<td>Finished Good</td>
<td>SCREWDRIVER</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>MYITEM02</td>
<td>Finished Good</td>
<td>HAMMER</td>
</tr>
<tr>
<td>3</td>
<td>01</td>
<td>MYITEM03</td>
<td>Finished Good</td>
<td>CHAINSAW</td>
</tr>
<tr>
<td>4</td>
<td>01</td>
<td>MYITEM04</td>
<td>Finished Good</td>
<td>CROWBAR</td>
</tr>
<tr>
<td>5</td>
<td>01</td>
<td>MYITEM05</td>
<td>Finished Good</td>
<td>WRENCH</td>
</tr>
<tr>
<td>6</td>
<td>01</td>
<td>MYITEM06</td>
<td>Finished Good</td>
<td>PLUNGER</td>
</tr>
<tr>
<td>7</td>
<td>01</td>
<td>MYITEM07</td>
<td>Finished Good</td>
<td>AWL</td>
</tr>
<tr>
<td>8</td>
<td>01</td>
<td>MYITEM08</td>
<td>Finished Good</td>
<td>PAINTBRUSH</td>
</tr>
</tbody>
</table>
Activity 5: Research the Advantage Link Import Resubmissions

Situation: The warehouse is having some issues whenever the handlers work with a particular item. In your research you identify some anomalies in the item master data and you want to find the root cause. As part of your investigation, you want to find all of the item imports (original and resubmissions) related to this item.

Procedures: In the previous activity you used the HighJump One Platform UI to correct and then manually resubmit a couple item import records – including MYITEM03. Additionally, you validated that the system successfully imported MYITEM03 into the item master table. However, Advantage Link also maintains a history of the resubmitted records. Now you will research the resubmitted item import records for MYITEM03 through the Advantage Link Administration pages.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Supply Chain Advantage | Advantage Link Administration | Resubmit Log menu.

The system displays the Search Resubmit Log window.

The only data in which you are interested is the item import data. You will include that as part of the filter on this page.

☐ Choose Import – Item in the Interface Type drop-down list.

☐ Click the Query button in the Action Bar.

The system shows all of the resubmitted item import batches that fall within the selected date window.
This page displays who resubmitted the batch; when it was resubmitted; the type of import; and the system-assigned identifier for the batch (Host Group ID). In the previous activity, you resubmitted three records, and the system created one new batch for each of them.

This page displays master information, but it does not allow you to view the actual data that was resubmitted for MYITEM03. You can view that information, but you must go to a different page.

☐ Expand the Advantage Link Administration | Imports nodes.

☐ Click the Item menu.

The system displays the Search Item Import page. This is the same page you used in a previous activity to view the original data sent from the host system to Advantage Link.

☐ Enter MYITEM03 in the Item Number edit box.

☐ Click the Query button in the Action Bar.

The system displays the Item Import window. This window displays all of the Advantage Link item import batches which include the MYITEM03 item number.
MYITEM03 was included in the original batch received from the host system along with four additional records. The original batch is noted in record number two on this window. (Note the count of 5 under the Master column.) Then you manually corrected and resubmitted MYITEM03. Advantage Link created a new item import batch that contained only that item. The resubmitted batch is noted in record number one on this window. (Note the count of 1 under the Master column.)

This window displays master information. If you want to see the actual data that was resubmitted you can drill into the details.

- Click the number 1 under the Master column for the second row (the resubmission)

The system displays the Item Master Import page for the specific batch.

This page includes the item import data elements that were included in the resubmitted batch. Note that the Warehouse ID value is 01 which is the value you manually entered in a previous activity. Also note that the Import Status is Success. This is consistent with the fact that you saw MYITEM03 in the item master table in a previous activity.
Activity 6: Create an Inventory Adjustment Export Record

**Situation:** The material handlers in the warehouse often perform Inventory Adjustments on the RF terminals. Behind the scenes, the Warehouse Advantage system passes the relevant information to the host system via Advantage Link for Flat File.

**Note:** Inventory Adjustment is a business process which allows a material handler to change the inventory quantity of an item in a location.

**Procedures:** In this activity you will run the Inventory Adjustment business process on the Virtual Terminal in order to create output which can be viewed through the Advantage Link Administration web pages. The emphasis of this exercise is on Advantage Link, so there are very few details on the various screens you will encounter on the Virtual Terminal.

- Start all solution environments in the AWESM tool, if necessary.
- Start the Virtual Terminal, if necessary.
- Log into the Virtual Terminal as AMY | AMY | FAAMY, if necessary.
- On the Virtual Terminal, press the F8 key in order to page down to the next set of options.
- Enter 7 (Inv Control) at the Option prompt.
- Enter 4 (Inv Adjustments) at the Option prompt.
- Enter 1 (Adjust by Loc) at the Option prompt.

The system displays the first prompt of the Adjust by Location business process.

```
Adjust by Loc
Scan the location.
LOCATION
```

In this example, you will change the inventory quantity of item NNFG07505 in location P101 to a quantity of 125.
☐ Enter **P101** at the **Location** prompt.

☐ Enter **NNFG07505** at the **Item Number** prompt.

☐ Enter **125** at the **Eaches (Quantity)** prompt.

The system displays the Reason Code prompt.

```
Adjust by Loc
Enter reason for adjustment.
REASON CODE
F8:LIST
```

The reason code is an identifier which indicates the specific reason the user changed the quantity of inventory. Perhaps it was because the inventory sustained some water damage. Perhaps it was because the inventory fell off the shelf and broke. In this case you will enter a reason code indicating it was related to a cycle count.

☐ Press the **F8 key** at the **Reason Code** prompt.

The system displays a list of eligible reason code values.

```
Adjust by Loc
1 07 Inv Adj CC
```

Each option consist of three parts: 1) a sequentially numbered value; 2) the unique reason code identifier; 3) a brief description of the reason code. In this example you will select Inv Adj CC which is choice number one, and it has a reason code of 07.

☐ Enter **1 (07 Inv Adj CC)** at the **Choice** prompt.
The system displays the Transaction Complete screen to indicate that you have successfully completed the Inventory Adjustment business process.

☐ Press the Enter key at the Transaction Complete screen.

☐ Press the F1 key at the Location prompt.

The system returns to the Inventory Adjustments menu screen.

Behind the scenes the system has passed the Inventory Adjustment data to Advantage Link. Advantage Link, in turn, has written the information to a text file which the host will then import and process. Next you will verify that the system has created the text file in the appropriate directory, and you will examine the contents of the text file.

☐ Open Windows Explorer, if necessary.

☐ Navigate to the D: \ ProgramData \ HighJump Software \ EXPORTS \ INVADJUST folder.

The system displays the contents of the folder.
The folder contains two files: a Windows Batch File and a text file with a DAT extension. When Advantage Link for Flat File receives a transaction from Warehouse Advantage it attempts to write the transaction to SCA_INVADJ.DAT. If that file already exists, then it writes the transaction to SCA_INVADJ.TMP and renames the file later. Next you will open the DAT file to validate that it contains the expected data.

- Open Notepad
- Choose the File | Open menu.
- Navigate to the D:\ProgramData\HighJump Software\EXPORTS\INVADJUST folder.
- Choose All Files in the drop-down list in the lower-right corner of the window.
- Click SCA_INVADJ.DAT in the upper portion of the window.
- Click the Open button.

The system opens the text file in Notepad.

As you review the data in Notepad you will find the Location (P101), the Item Number (NNFG07505), the Quantity (125), and the Reason Code (07). The presence of this file and the data in it indicates that Advantage Link for Flat File successfully exported the data to the host.

In a production environment, the host system will delete the file after it has processed it. More than likely, the host will keep a limited number of backup files which can also be reviewed during the debugging process if the DAT file no longer exists. The reason the SCA_INVADJ.DAT file exists here in the training environment is because there are no host systems installed on the training environments.

- Choose the File | Exit menu.

In the next activity, you will research the Advantage Link export results.
Activity 7: Research the Advantage Link Export Results

**Situation:** The host system has not received any transactions from the HighJump system in the last 30 minutes. There are several reasons this might occur. You decide to investigate the HighJump system first to ensure that it is sending the transactions correctly.

**Procedures:** In the previous activity you performed an Inventory Adjustment transaction on the Virtual Terminal. The Warehouse Advantage system is designed to send the Inventory Adjustment transaction information to Advantage Link for Flat File which then sends it to the host via a text file. In this activity you will spot check the data in the Advantage Link Administration web pages in order to validate that HighJump’s role in the data transmission is working as intended.

- Open the HighJump One Platform UI, if necessary.
- Click the Menu | Supply Chain Advantage | Advantage Link Administration | Exports | Inventory Adjustment menu.

The system displays the Search Inventory Adjustment Export window.

This window allows you to enter filter criteria for viewing the various inventory adjustment export transactions in Advantage Link. In this example the data set is small, so there is no need to enter any filtering elements.

- Click the **Query** button in the **Action Bar**.

The system displays the Inventory Adjustment Export window.
There is only one record in this window. It reflects the Inventory Adjustment transaction you performed on the Virtual Terminal in the previous activity. The chart below describes each one of the columns on the web page.

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Group ID</td>
<td>A system-assigned identifier.</td>
</tr>
<tr>
<td>Record Create Date</td>
<td>The date the system exports the transactions.</td>
</tr>
<tr>
<td>Master</td>
<td>The number of inventory adjustment transactions included in the export.</td>
</tr>
<tr>
<td>Resubmit</td>
<td>Allows you to export the data a second time. (More on this later.)</td>
</tr>
</tbody>
</table>

In addition to viewing the Host Group ID and the export date, you can also see the inventory adjustment data that was received from Warehouse Advantage (and which is passed to the host.) However, you first must navigate to a different web page.

☐ Click the number 1 under the Master column.

The system displays the Inventory Adjustment Master Export window.

This screen displays the Advantage Link data received from Warehouse Advantage.

☐ Use the horizontal scroll bar to review the Item Number, From Location ID, Quantity After, and the Reason Code elements.
Note that each of the elements corresponds with the data you entered on the Virtual Terminal screen.

<table>
<thead>
<tr>
<th>Virtual Terminal</th>
<th>HighJump One Platform UI</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>From Location ID</td>
<td>P101</td>
</tr>
<tr>
<td>Item ID</td>
<td>Item Number</td>
<td>NNFG07505</td>
</tr>
<tr>
<td>Eaches</td>
<td>Quantity After</td>
<td>125</td>
</tr>
<tr>
<td>Reason Code</td>
<td>Reason Code</td>
<td>07 (Inv Adj CC)</td>
</tr>
</tbody>
</table>

The presence of the transaction on this screen indicates that Advantage Link for Flat File received the transaction from WA and that it attempted to send the transaction to the host system. The research on the text file you performed in the previous activity confirms that Advantage Link for Flat File successfully transmitted the text file to the host.

In the next activity you will see how to modify that data in an export, with the intent of resubmitting it to the host system.
Activity 8: Modify the Advantage Link Export Data

Situation: Several of the Inventory Adjustment transactions that Warehouse Advantage sends to the host are failing at the host because they contain references to invalid reason codes. Rather than correct the transactions at the host level, you opt to correct them in the HighJump system and then resubmit them to the host for processing.

Procedures: In the previous activity you performed an Inventory Adjustment transaction on a Virtual Terminal. Then you reviewed the transaction in the Advantage Link Administration web pages and in the text file created for the host. In this activity you will modify the reason code portion of the Inventory Adjustment transactional data through the Advantage Link Administration web pages. Then in a later activity you will resubmit the transaction to the host.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Supply Chain Advantage | Advantage Link Administration | Exports | Inventory Adjustment menu.

☐ Click the Query button in the Action Bar.

The system displays the Search Inventory Adjustment Export window.

☐ Click the Query button in the Action Bar.

The system displays the Inventory Adjustment Export window.
This page indicates that there is one batch (or group) of records submitted to the host that meets the search criteria provided on the previous page. The 1 under the master column indicates that there is exactly 1 transaction included in this batch. You can see the details of the 1 transaction, but first you must navigate to a different page.

☐ Click the number 1 under the **Master** column.

The system displays the Inventory Adjustment Master Export window.

This screen displays the Advantage Link data received from Warehouse Advantage.

☐ Use the **horizontal scroll bar** to review the **Reason Code** element.

The system indicates that it passed a reason code of 07 to the host as part of this transaction.

The system allows you to change this value, but you first must navigate to another window.
☐ Click the number 1 in the first column on the left.

The system displays all attributes of the exported record, and it gives you the opportunity to change them.

☐ Enter 88 in the Reason Code edit box.

☐ Click the Update button in the Action Bar.

The system returns to the Inventory Adjustment Master Export window.

☐ Use the horizontal scroll bar to review the Reason Code element.

The system indicates that the reason code now has a value of 88.
In this activity you utilized the Advantage Link Administration web pages to change the reason code value of an Inventory Adjustment export transaction. However, the new data has not yet been sent to the host. Your change was limited to Advantage Link. In the next activity you will learn how to resubmit this new data to the host.
Activity 9: Resubmit an Advantage Link Export Batch

Situation: Several of the Inventory Adjustment transactions that Warehouse Advantage sends to the host are failing at the host because they contain references to invalid reason codes. Rather than correct the transactions at the host level, you opt to correct them in the HighJump system and then resubmit them to the host for processing.

Procedures: In the previous activity you utilized the Advantage Link Administration web pages to change the reason code value of an Inventory Adjustment export transaction. In this activity you will resubmit this new data to the host.

☐ Open the HighJump One Platform UI, if necessary.

☐ Click the Menu | Supply Chain Advantage | Advantage Link Administration | Exports | Inventory Adjustment menu.

☐ Click the Query button in the Action Bar.

The system displays the Search Inventory Adjustment Export window.

![Search Inventory Adjustment Export window](image)

☐ Click the Query button in the Action Bar.

The system displays the Inventory Adjustment Export window.

![Inventory Adjustment Export window](image)
When working with imports, you can resubmit at the batch level or at the individual record level. However, when you work with exports, the only option for resubmitting is at the batch level. If you resubmit a batch, then the system resubmits all transactions included in that batch. In this specific example, the batch (group) contains exactly one transaction, so submitting the batch will resubmit that one transaction.

☐ Click the **Resubmit** link for the record.

The system resubmits the entire contents of the batch (1 transaction) and then displays a screen indicating that it has taken place.

You corrected the invalid reason code in the Inventory Adjustment transaction. Then you resubmitted the entire batch. Now you will verify that the system successfully sent the modified transaction to the host via a text file.

☐ Open **Windows Explorer**, if necessary.

☐ Navigate to the D:\**ProgramData** \**HighJump Software** \**EXPORTS** \**INVADJUST** folder.

The system displays the contents of the folder.
The folder contains three files: a Windows Batch File, a text file with a DAT extension, and a text file with a TMP extension. When Advantage Link for Flat File receives a transaction from Warehouse Advantage it attempts to write the transaction to SCA_INVADJ.DAT. If that file already exists, then it writes the transaction to SCA_INVADJ.TMP and renames the file later. In this case, the DAT file contains the original export data. When you resubmitted the data, the system recognized that the DAT file existed, so it wrote the transaction to the TMP file. Next you will open the TMP file to validate that it contains the expected resubmitted data.

☐ Open Notepad

☐ Choose the File | Open menu.

☐ Navigate to the D: \ ProgramData \ HighJump Software \ EXPORTS \ INVADJUST folder.

☐ Choose All Files in the drop-down list in the lower-right corner of the window.

☐ Click SCA_INVADJ.TMP in the upper portion of the window.

☐ Click the Open button.

The system opens the text file in Notepad.

<table>
<thead>
<tr>
<th>SCA_INVADJ.TMP - Notepad</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Edit Format View Help</td>
</tr>
<tr>
<td>851 NNFG07505 NNFG07505</td>
</tr>
</tbody>
</table>

☐ Use the horizontal scroll bar to review the data.

As you review the data in Notepad you will find the modified reason code value of 88. The presence of this file and the data in it indicates that Advantage Link for Flat File successfully exported the resubmitted data to the host.

<table>
<thead>
<tr>
<th>SCA_INVADJ.TMP - Notepad</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Edit Format View Help</td>
</tr>
<tr>
<td>AVAILABLE AVAILABLE 88 08/03/15P101</td>
</tr>
</tbody>
</table>

The reason the TMP file and the DAT file still exist in the folder structure is because there is no host system installed in the training environment. In a production environment, the host system would eventually process and delete these files. If the host system maintains backups of the text files, then you may be able to review those backup files as you conduct your research.

☐ Choose the File | Exit menu.
As a result of the resubmission, the system sent the modified Inventory Adjustment transaction to the host via a text file. Additionally, it created some history about the manual resubmission. In the next activity you will perform some research around the history of the resubmission.
Activity 10: Research the Advantage Link Export Resubmissions

**Situation:** The host is still having some issues regarding the Inventory Adjustment transactions it receives from the HighJump system. The warehouse is having some issues whenever the handlers work with a particular item. In your research you identify some anomalies in the item master data and you want to find the root cause. As part of your investigation, you want to find all of the item imports (original and resubmissions) related to this item.

**Procedures:** In the previous activity you used Advantage Link Administration web pages to correct and then manually resubmit a couple item import records – including MYITEM03. Additionally, you validated that the system successfully imported MYITEM03 into the item master table. However, Advantage Link also maintains a history of the resubmitted records. Now you will research the resubmitted item import records for MYITEM03 through the Advantage Link Administration pages.

- Open the HighJump One Platform UI, if necessary.
- Click the Menu | Supply Chain Advantage | Advantage Link Administration | Resubmit Log menu.

The system displays the Search Resubmit Log window.

![Search Resubmit Log](image)

The only data in which you are interested is the data related to the export of Inventory Adjustments. You will include that as part of the filter on this page.

- Choose **Export – Inventory Adjustment** in the **Interface Type** drop-down list.
- Click the **Query** button in the **Action Bar**.

The system shows all of the resubmitted export batches of Inventory Adjustment transactions that fall within the selected date window.
This page displays who resubmitted the batch; when it was resubmitted; the type of export; and the system-assigned identifier for the batch (Host Group ID). In a previous activity, you resubmitted exactly one Inventory Adjustment transaction, and the system reflects that here.

This concludes the exercise on the Advantage Link Administration application. By using the skills you learned in the exercise, you will be able review, modify, and resubmit the data exchanged between Warehouse Advantage and a host system.
Admin Lab Exercise 11: Advantage Architect Applications

Introduction

This exercise consists of the following activities:

1. Export an Advantage Architect Application
2. Import an Advantage Architect Application
3. Compile and Activate an Advantage Architect Application
4. Test the Changes
5. Restore the Original Application

Background Information

Advantage Architect is the development tool created by HighJump Software which is used to modify the business processes that run on the RF terminals.

Most HighJump customers have three fully functional instances of the HighJump system. One instance is the system of record. This instance is called the “live” environment or the production environment. The second instance is a development environment where changes can be developed. The third instance is a test environment where system testing takes place.

Making changes to the HighJump system is standard practice in most warehouses. Sometimes the end users ask for enhancements; sometimes a business process does not work according to the design; and sometimes the warehouse operation changes. All of these cases demand a change to the HighJump system.

These changes are typically made in the development environment. Then those changes are copied to the test environment where a different team validates that they work as expected. Then, lastly, the changes are copied to the production environment. The process of copying changes from one environment to another is called a “promotion” or a “release”. As far as Advantage Architect is concerned, a promotion involves exporting the Architect application out of one environment and then importing / compiling / activating it in another environment.

HighJump has a recommended methodology for managing releases and promotions. The recommendations are documented in a “Best Practice” file in the desktop folder. You can also obtain the document by contacting the HighJump Support Team.
Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Architect Apps – Import and Export menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Export an Advantage Architect Application

Situation: The daily operations in the warehouse have changed, and the warehouse manager has requested a handful of changes to the business processes. Your team has made the corresponding changes to Advantage Architect in the development environment. Now you are ready to promote those changes to a test environment so that the warehouse supervisors can test the new functionality.

Procedures: The promotion process involves an export out of the development environment and an import / compile / activate into the test environment. The procedures below demonstrate how to export the WA Advantage Architect application as part of a promotion process. (They do not demonstrate how to import the application into the next environment.)

First, you will create a folder structure on the desktop which will hold the exported application. In the real-world, this folder would likely be on the network. Since this scenario is in the context of promoting an application from one environment to another, you will create a folder name that contains a sequential release number. The folder structure you create will look like similar to the following screenshot.

☐ Right-click anywhere on the Desktop.

☐ Choose the New | Folder menu option.

☐ Enter My Releases in the edit box for the name of the folder.

☐ Double click the My Releases folder you just created in order to open it.

☐ Right click anywhere in the panel on the right.

☐ Choose the New | Folder menu option.

☐ Enter Release 01 in the edit box for the name of the folder.

The folder structure should look like the following snapshot.
The folder is created. Next you will launch the Advantage Architect tool and log into it.

- Choose the Start | All Programs | HighJump Software | Advantage Architect menu option.

The system opens the Connect to Application Repository window. A repository is a SQL Server / Oracle database that houses all of the Advantage Architect objects for the Advantage Architect applications. In this window you instruct the system where the repository is located.

- Enter (local) in the Server Name edit box.
- Enter REPOS in the Database Name edit box.
- Select the Use SQL Server Authentication radio button.
- Use the Architect section of the login.txt file in the folder on the desktop to populate the Login and Password edit boxes of the Connection dialog.
- Click the Connect button.

The system displays the Application Login window.
☐ Type your name in the **User Name** edit box. (There is no validation on this entry.)

☐ Click the **OK** button.

The system opens the Advantage Architect tool.

Next you will export the WA application from Advantage Architect into the folder you created above.

☐ Choose the **File | Export | Export Application** menu.

The system opens the Export Application window.
☐ Choose **WA** in the **Application** drop-down list.

☐ Click the **ellipsis button** to the right of the **Export Folder** edit box.

☐ Navigate to the **Desktop \ My Releases \ Release 01** folder.

☐ Click the **OK** button.

The Export Application window should look like the following graphic.

![Export Application Window](image)

☐ Click the **OK** button.

As Advantage Architect exports the application, it displays the status in the Output Window near the bottom of the screen.

```
Exporting WA to C:\Users\HJAccount\Desktop\My Releases\Release 01
Exporting Application Data to XML ...
Exporting Process Object Data to XML ...
Exporting Calculate Action Data to XML ...
Exporting Compare Action Data to XML ...
```

When the export is completed successfully, the system displays the Completed dialog.

![Completed Dialog](image)
☐ Click the OK button.

☐ In Windows Explorer, view the contents of the Desktop \ My Releases \ Release 01 folder.

The system has created a set of XML files in the folder.

At this point, you could leave the XML files in their current state. However, it is often easier to deal with a single ZIP file instead of multiple XML files. This is especially true if you need to promote an Advantage Architect application from one environment to another. Follow the procedures below to zip the XML files into a single ZIP file.

☐ Press Control + A to select all of the files in the Release 01 folder

☐ Right-click any of the selected files.

☐ Choose the Send To | Compressed (Zipped) Folder menu option.

The system creates a zipped file amongst the other XML files. The name of your zipped file may be different than the one in the diagram.

☐ Select all of the files in the folder EXCEPT for the zipped file you just created.

☐ Press the Delete key on the keyboard.

The system presents a window confirming that you want to delete the files.
Click the *Yes* button.

The system deletes all of the XML files and leaves the zip file in the folder. The name of your zipped file may be different than the one in the diagram.

The name of the zip file should indicate the application name (WA) and the release number (release 01). This is to prevent any confusion if the zip file becomes separated from the actual release folder.

Rename the ZIP file to `wa_architect_release_01.zip`

The contents of the folder should look like the following screenshot.

Every release folder should include a document that describes the changes included in the release. This document is called the release notes. Usually, the developers who make the changes will create the document. The “Best Practices” document contains a recommended template for the release notes. In this example, you will create a very simple set of release notes.

Open Notepad

Add the following lines (or something similar) to the new document.

```
=================================
RELEASE 01
=================================

1. Contains the WA base training application.
```
Notepad should look like the following graphic.

```
FILE EDIT FORMAT VIEW HELP

RELEASE 01

1. Contains the WA base training application.
```

☐ Choose the **File | Save** menu option.

☐ Click the **Desktop** link in the panel on the left.

☐ Navigate to the **My Releases | Release 01** folder.

☐ Enter *release_notes_for_release_01* in the **File Name** edit box.

☐ Click the **Save** button.

☐ Choose the **File | Exit** menu option.

The system creates the release notes in the same folder as the zipped WA application.

In this activity you exported the WA application into a release folder and you created a corresponding set of release notes. At this point, the release folder contains all of the necessary files, and it is ready for promotion into the next environment.
Activity 2: Import an Advantage Architect Application

Situation: The daily operations in the warehouse have changed once again, and the warehouse manager has requested a handful of changes to the business processes. Due to the time-sensitive nature of the changes, your company opts to contract with the HighJump Software Professional Services team to make the changes. The HighJump Software Professional Services team has developed and tested the changes on their servers in Minneapolis, MN, and they have determined that the changes meet the warehouse manager’s requirements. They have exported the WA Advantage Architect application (from the Minneapolis servers). They have zipped the exported WA application into a single zipped file, and they have placed the zipped file on an FTP site along with a set of release notes. It is your responsibility to grab the files, and then promote the WA application into your development environment.

Among the many changes requested by the warehouse manager, he has requested that the EQUIPMENT/ZONE prompt included in the Logon process be changed to read FORKLIFT.

Procedures: The promotion process involves an export out of the development environment and an import / compile / activate into the test environment. The procedures below demonstrate how to import the WA Advantage Architect application as part of a promotion process. (They do not demonstrate how to export the application as that was covered in a previous activity.) After the promotion, the EQUIPMENT/ZONE screen will read FORKLIFT.

Current State: Like all other activities in this manual, you will examine the current state of the process before you introduce any changes.

- Log in to the Virtual Terminal with a user and password.

The system displays a prompt for EQUIPMENT/ZONE. The prompt title on this screen will read FORKLIFT after the promotion is complete.
If the HighJump team performs the development on the servers Minneapolis, they may deliver the Advantage Architect application to you via an FTP site or a document sharing site. You will need to download the application and then place it in some sort of release folder structure similar to the one you built in a previous activity. In this example, the instructor has already “downloaded” the files and placed them into the folder structure on your desktop.

☐ Open the Training folder on the desktop.

☐ Navigate to the Releases \ Release 09 ... folder.

The folder contains a set of release notes and a zipped Advantage Architect application. The folder should look like the following graphic.

Because the Advantage Architect application is zipped, you will need to unzip it before you can import it into the tool.

☐ Right click the wa_architect_release_09 file.

☐ Choose the Extract All menu option

The system displays the Extract Compressed (Zipped) Folders window.

The edit box contains the default folder where the system will place the extracted files. There is no need change the value, but you will reference this folder when you later import the application into Advantage Architect.
☐ Click the Extract button.

The system creates an unzipped folder called wa_architect_release_09.

The system automatically navigates to the folder containing the extracted files. It displays a list of the unzipped XML files which contain the entire WA Advantage Architect application.

The files are extracted, and now you can import them into Advantage Architect.

☐ Open the Advantage Architect tool and log into it, if necessary.

☐ Choose the File | Import | Import Application menu.

The system displays the Import Application window.

☐ Click the ellipsis button to the right of the Application edit box.

☐ Navigate to the Training \ Releases \ Release 09 ... \ wa_architect_release_09 folder.

☐ Click the OK button.
The system updates the Application edit box with the selected folder.

![Import Application dialog box]

**WARNING:** The directory path of the Import Application defaults to the previously completed import. Selecting a directory path other than the one specified in the activity steps will result in a different application being promoted.

☐ Click the **OK** button.

Because the WA application already exists in the repository, the system displays the Application Conflict dialog. The “Overwrite” option will delete the existing WA application from Advantage Architect and then import the new one. The “Merge” option will create a blended application using both of them. In a previous activity you exported the WA application, and that export will serve as the backup. Because you have a backup application, you don’t need to be concerned with the “Overwrite” option deleting the existing WA application. In this example you will use the “Overwrite” option.

![Import Application dialog box: Overwrite option selected]

☐ Select the **Overwrite the existing application** radio button.

☐ Click the **OK** button.

The system first deletes the WA application in the repository. Then it imports the XML files into the repository. The system displays the progress in the output window.
When the import is complete, the system displays the Completed dialog.

☐ Click the OK button.

Importing the application is not enough to impact the business processes running on the terminals. In order for the terminals to reflect the changes, you need to compile and activate the application. That is the content of the next activity.
Activity 3: Compile and Activate the Changes

In the activity above you imported a new Warehouse Advantage Architect application in order to, among other things, to change the EQUIPMENT/ZONE prompt to FORKLIFT. However, if you view the Virtual Terminal, you will see that the changes have had no impact. In order to see the changes on the Virtual Terminal, you must first compile the application and then activate the application (to the Virtual Terminals).

The compile component looks for syntax errors in the application. If it finds no syntax errors, then it creates a single file on the hard drive which contains all of the business rules for the given application. The activation component makes this modified application available to all of the RF terminals. The next several procedures demonstrate how to compile and activate the WA application.

☐ Choose the Run | Compile Application menu.

The system displays the Compile Application dialog.

☐ Choose WA in the Application drop-down list.

The system automatically populates the Compilation Folder edit box. There is no need to change it.

☐ Select the Production Compile radio button.

☐ Click the Compile button.

The system displays several informational messages related to the compile in the Output Window.
It also displays a progression bar in lower left corner of the tool to indicate that the compile is in progress.

When the compile is complete the system displays a dialog indicating that it was successful along with the number of warnings that it found. In this case, the warnings are because some of the modules (Labor Advantage, Yard Advantage) have not been installed in the training environment. These warnings will not cause any problems with the Warehouse Advantage application.

At this point the system gives you an option for activating the application.

☐ Click the Yes button.

The system displays the Activate Application dialog.
The Compilation Folder edit box points to the folder which contains the output from the compile. The system defaults this value correctly. There is no need to change it.

The Solution Name setting is related to the application name. Since you are working with the WA application, the solution name needs to match.

☐ Select WA in the **Solution Name** drop down box.

The radio buttons in the Activation Options section indicate how the engine should behave during the activation process.

☐ Select the **Stop and Restart** radio button in the **Activation Options** section.

**Note:** The Dynamically Activate option is a feature which has not been fully implemented yet.

In order to activate the WA application you must shut down its solution environment. However, if you choose to shut down the solution environment, the system will automatically kick off all active users from the RF terminals. If an RF terminal user is in the middle of a picking process, the system will automatically kick him out of that process. And if that happens, it is likely that the user will have some problems when he logs back into the RF terminal. For this reason, it is important that you verify nobody is actively running a business process on an RF terminal before you shut down the solution environment.

☐ On the **Virtual Terminal**, navigate back to the **USER ID** screen, if necessary.

☐ In **Advantage Architect**, click the **OK** button.

Behind the scenes, the system shuts down the WA solution environment; it moves a couple files to a different folder; and it updates a couple database tables. Once that is complete, the activation process restarts the WA solution environment. When the activation process is finished, the system displays a confirmation window.
☐ Click the **OK** button.

The compile and activation process is complete. Next you will test the changes.
Activity 4: Test the Changes

You have imported a new WA application into Advantage Architect. And you have compiled and activated the application. The next step in the process is to validate that the changes work according to the specified requirements.

The chart below outlines the various test cases that must be validated. Each test case defines an action you must take, and it defines the expected outcome based on the requirements. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

The test cases do not detail every keystroke. You need to use the knowledge you gained in the first activity of this exercise in order to test the scenarios.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Outcome</th>
<th>Pass / Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start the Virtual Terminal and log in with a <strong>user</strong> and <strong>password</strong>.</td>
<td>The system displays a screen with a FORKLIFT prompt title.</td>
<td></td>
</tr>
</tbody>
</table>

The presence of the FORKLIFT prompt indicates that you have successfully promoted the WA application into the environment. At this point, the testing team could test the other changes introduced into the system by the HighJump Software Professional Services team.
Activity 5: Restore the Original Application

Background: In the previous activities you replaced the WA Advantage Architect application with a different one. In order to continue with the remainder of the exercises, you need to restore the WA Advantage Architect application back to its state prior to this exercise.

Procedures: The procedures below demonstrate how to restore the original WA application.

At the beginning of this exercise you exported the WA application into the Desktop \ My Releases \ Release 01 folder with the intent of promoting it to another environment. Then you zipped the exported XMLs into a single file. That zipped file holds the original WA Advantage Architect application, and that zipped file is the one that you will import back into Advantage Architect.

☐ Open the My Releases folder that is on the desktop

☐ Open the Release 01 folder.

The contents of the folder look like the following graphic.

☐ Right click the wa_architect_release_01 file.

☐ Choose the Extract All menu option

The system displays the Extract Compressed (Zipped) Folders window.
Admin Lab Exercise 11:
Advantage Architect Applications

The edit box contains the default folder where the system will place the extracted files. There is no need change the value, but you will reference this folder when you later import the application into Advantage Architect.

☐ Click the **Extract** button.

The system creates an unzipped folder which holds several XML files containing the entire WA Advantage Architect application.

☐ Click the **Extract** button.

The files are extracted, and now you can import them into Advantage Architect.

☐ Open the **Advantage Architect** tool and log into it, if necessary.

☐ Choose the **File | Import | Import Application** menu.

The system displays the Import Application window.

☐ Click the **ellipsis** button to the right of the **Application** edit box.

☐ Navigate to the **My Releases \ Release 01 \ wa_architect_release_01** folder.

☐ Click the **OK** button.

The system updates the Application edit box with the selected folder.
Click the **OK** button.

Because the WA application already exists in the repository, the system displays the Application Conflict dialog. The “Overwrite” option will delete the existing WA application from Advantage Architect and then import the new one. The “Merge” option will create a blended application using both of them. In this example you will use the “Overwrite” option.

Select the **Overwrite the existing application** radio button.

Click the **OK** button.

The system first deletes the WA application in the repository. Then it imports the XML files into the repository. The system displays the progress in the output window.
When the import is complete, the system displays the Completed dialog.

- Click the **OK** button.

Importing the application is not enough to impact the business processes running on the terminals. In order for the terminals to reflect the changes, you need to compile and activate the application.

- Compile and activate the **WA** application (see the previous activities for additional instructions.)

The compile and activate procedures complete the restoration process. The next step is to validate that the imported application reflects the original version of the WA application. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps in this activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
</table>

Admin Lab Exercise 11: Advantage Architect Applications  245
Log into the **Virtual Terminal** with a **user** and a **password**.

The system displays a prompt for **EQUIPMENT\ZONE**.

If this test condition passed, then you have successfully restored the WA application back to the way it was prior to this exercise. You can continue with the next exercise.
Admin Lab Exercise 12:
Visual Debugger

Introduction
This exercise consists of the following activities:
1. Understand the Current Process
2. Explore the Visual Debugger
3. Add a Watch to Visual Debugger
4. Add a Breakpoint to Visual Debugger
5. Use the Message Watch Feature of Visual Debugger
6. Use the Record Feature of Visual Debugger
7. View the Call Stack
8. View a List

Background Information
Visual Debugger is a tool that allows you to troubleshoot the business logic (workflows) of your Advantage applications. Technically speaking, Visual Debugger is a tool that allows you to troubleshoot the process objects that were created in Advantage Architect.

The concept of a process object is described in great detail during the Advantage Architect class. For the purposes of this class, a process object is defined as “a set of executable steps designed to accomplish a specific purpose.” They are created in a development tool called Advantage Architect. For example, one process object might prompt the user for an order number. Another process object might retrieve all order master data from the database for a given order number. Each step in the process object returns either a PASS or FAIL status.

Developers use this tool extensively to debug logic problems in the application. However, as an administrator you will use it primarily under the direction of the HighJump support team. They may ask you to capture some information and send it to them via e-mail. Or they may ask you to run the Visual Debugger and then explain what you see in certain windows.

Regardless of the mode you use when you run Visual Debugger, it will always have a negative impact on the performance of the terminal to which it is attached. It is a good idea to set that expectation with the user who is running the terminal.
Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the Debugger menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Understand the Current Process

This activity demonstrates how the Location Status process operates in the Warehouse Advantage base application. This business process will be used as a foundation for learning about the Visual Debugger tool. The Location Status process manages the viewing and/or modification of a specific location's status. The user is first prompted to scan the location to be viewed or modified. The location's current status is then displayed and the user can enter a new status. The possible location status codes are listed below.

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Empty location, where there is no inventory associated to the location.</td>
</tr>
<tr>
<td>P</td>
<td>Partial location, where there is at least one item in the location and the location has not reached a user-determined capacity.</td>
</tr>
<tr>
<td>F</td>
<td>Full location, where there is at least one item in the location and the location has reached a user-determined capacity.</td>
</tr>
<tr>
<td>I</td>
<td>Inactive location, where the location and associated inventory are not available for put-away and picking.</td>
</tr>
</tbody>
</table>

After a new status is entered by the user the system updates the location's status with the identified status and a location status transaction is generated.

- Log in to the Virtual Terminal as AMY / AMY / FAAMY.
- Press the F8 key to scroll down a page.
- Enter option 7 (the Inv Control menu.)
- Enter option 3 (the Inventory Status menu.)
- Enter option 4 (the Location Status menu.)

The system displays the Location screen.

- Enter I171 at the Location dialog.
The system displays the location and the current status (empty) at the top of the location status screen.

Enter I at the Location Status dialog.

The system updates the status of the location in the location master table, and then loops back to the top of the business process. The system now gives the user an opportunity to view or modify the status of another location.

Enter I171 at the Location dialog.

In this example, you will set the status of I171 back to empty.

Enter I171 at the Location dialog.

The system displays the location and the current status (inactive) at the top of the location status screen.
Enter \textbf{E} at the \textbf{Location Status} dialog.

The system updates the status of the location in the location master table, and then loops back to the top of the business process. The system now gives the user an opportunity to view or modify the status of another location.

Press the \textbf{F1 key} to navigate back to the Inventory Status menu.

Press the \textbf{F1 key} to return to the Inv Control menu.

Press the \textbf{F1 key} to return to the main menu.
Activity 2: Explore the Visual Debugger

Scenario: You recently contracted with HighJump Software to make some changes to the Location Status business process. After promoting the change into production, one of your managers indicates that the data the HighJump system is sending to the host system for this business process is no longer correct. You call the HighJump support team, and they ask you to start the Visual Debugger, walk through a couple steps line by line, and then verbally describe what you see.

Procedures: The procedures below demonstrate how to start the Visual Debugger and how to step through each line of a process objects. However, they do not reproduce the problem with the incorrect data going to the host system.

☐ Configure the Device Name parameter of the Virtual Terminal to RADIO_01, if necessary.

☐ Navigate to the Location prompt of the Location Status business process.

☐ Choose the Start | All Programs | HighJump Software | Visual Debugger menu.

The system displays the main screen of the Visual Debugger. In the initial state, the window is empty. The Visual Debugger window will remain empty until you connect it to a specific application server and a specific device.

☐ Choose the File | Connect to Application Server menu.

The system displays a dialog for entering the connection information to the application server.
In this specific example, the Visual Debugger resides on the same machine as the application server. This is defined as a local connection. The system also provides two additional methods (IPC and TCP/IP) for connecting to a remote application server. For additional information on these other methods, see the Visual Debugger help files.

☐ Choose the **Local** radio button.

☐ Click the **OK** button.

The system changes the title bar of the window to indicate that it successfully connected to the application server. However, it is still not attached to a specific device, and the main window remains empty.

☐ Choose the **File | Select Device** menu.

The system displays a drop-down list of all possible devices that can be debugged. The list includes Virtual Terminals, Web Terminals, physical RF terminals, as well as other items that run process objects.
Choose **RADIO_01** in the **Select Device** drop-down list.

Click the **OK** button.

The main window of the Visual Debugger remains empty. However, the title bar now reflects the device, the server, and the port number.

One of the key concepts of Visual Debugger is the concept of control. Either Visual Debugger has control of a process object or it is waiting for input from the end user. If Visual Debugger is not responding to your mouse clicks, it is likely that it is waiting for the end user to enter or scan a piece of data. In the current state, Visual Debugger is connected to a device, but you cannot do anything in the tool, because it is waiting for you to enter something at the Location prompt in the Virtual Terminal.

Enter **M105** at the **Location** dialog of the Virtual Terminal.

The system displays several lines of data in the Visual Debugger tool. These are the lines of the process object.
At this point Visual Debugger has control of the process, and it is waiting for you to step through it line by line. Additionally, the Virtual Terminal remains on the LOCATION dialog. If the device was a physical RF radio, then the screen would eventually read “Advantage Engine Down.” The yellow arrow points to the line number which the system will execute next. If you want to execute that one line, then you can use the “Step Into” feature.

☐ Choose the Debug | Step Into menu in the Visual Debugger.

The system executes that one line; advances the yellow arrow to the next line that will be executed; and then Visual Debugger stops. In this case, the process object also changes so the display in the top panel of Visual Debugger also changes.

This second button from the left is the equivalent of the Debug | Step Into menu. For the purpose of this document, it will be referred to as the Step Into button.

☐ Click the second button from the left in the Visual Debugger.
Once again, the system executes that one line; advances the yellow arrow to the next line that will be executed; and then Visual Debugger stops. In this case, the process object also changes so the display in the top panel of Visual Debugger also changes.

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Pass</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Calculate : ScrTxt: Enter Location</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Call : Dialog</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Compare : <a href="">F1:Cancel</a>&gt;?</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Compare : Dialog Field = &quot;&quot;?</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Calculate : Location = Dialog Field</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Call : Verify Location</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Compare : LOC Type = &quot;F&quot;?</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

- Click the **Messages** tab along the bottom of Visual Debugger.

- Click the **Step Into** button of Visual Debugger multiple times until the Virtual Terminal displays the **Location Status** dialog.

As you step through the process object lines, the system displays each executed line in the bottom pane along with the returned status of each line (PASSED or FAILED). A status of FAILED does not necessarily mean that something is wrong. Often times a FAILED status is the expected result.

- Click the **Step Into** button of Visual Debugger one more time.

The system displays the following warning. The text of the warning indicates that Visual Debugger is waiting for input at the Virtual Terminal. The fix is to acknowledge the warning, and then enter data at the terminal.
☐ Click the **OK** button.

☐ Enter **P** at the **Location Status** dialog of the Virtual Terminal.

At this point, you can step to the appropriate line as directed by the HighJump support team, and then read them the information you see on the screen. Once they have the necessary information, you need to exit the debugger which will return the terminal back to normal operation.

☐ Click the **Step Into** button of Visual Debugger a couple more times.

You are done utilizing the Step Into feature, so now you will exit the tool.

☐ Choose the **File | Detach Debugger** menu in Visual Debugger.

The system closes the Visual Debugger. The Virtual Terminal advances to next prompt. And Virtual Terminal returns to normal operation. There may be times when you choose the File | Detach Debugger menu and the system does nothing. This is most likely because the debugger is waiting for input from the user. If this is the case, then enter data at the Virtual Terminal, and then the debugger will disappear.

☐ In Virtual Terminal, navigate back to the **menu**.
Activity 3: Add a Watch

**Background:** The Visual Debugger tool provides the ability to look at the value of a variable in memory. This is an important aspect of the troubleshooting process as it allows you to see exactly which line modifies a given variable. The concept of looking at the value of a variable through the lens of Visual Debugger is called a “Watch”.

**Scenario:** This scenario builds upon the one from the previous activity. The HighJump support team has reviewed the data you read to them, but they have not found the root cause of the problem. They want you to set a Watch on the Location variable, the Status variable, and the Transaction Code variable in order to assist with the troubleshooting process.

**Procedures:** The procedures below demonstrate how to add a Watch to Visual Debugger. However, they do not reproduce the problem with the incorrect data going to the host system.

- Open the Virtual Terminal, if necessary.
- Navigate to the Location prompt of the Location Status business process.
- Open the Visual Debugger.
- Attach the Visual Debugger to the Virtual Terminal.
- Enter M106 at the Location prompt of the Virtual Terminal.

The system displays several lines of the current process object.

```
29:Compare  : SCR HELP3 = ""? Pass: 31 Fail: 30
30:Dialog   : PRMT-1:OPT-3:LINE-N Pass: 37 Fail: 30
33:Compare  : SCR HELP2 = ""? Pass: 35 Fail: 34
  37:Return  : PASS        Pass: 0 Fail: 0
  38:Return  : FAIL        Pass: -1 Fail: -1
```

- Click the Watch Fields tab along the bottom of the Visual Debugger.

The system displays a list of all of the currently defined Watch variables in the lower pane. In this case, there are none.
Choose the Watch | Add Field menu.

The system displays the Add Watch window.

Choose Location in the Fields in WA drop-down list.

Click the OK button.

The system adds the Location variable to the lower pane, and it displays the current value of the Location variable. In this case, the current value is empty.

In this example you will watch two additional fields: Status and Transaction Code.

Choose the Watch | Add Field menu.

Choose Status in the Fields in WA drop-down list.

Click the OK button.
The system adds the Status variable to the lower pane, and it displays the current value of the Status variable. In this case, the current value is empty.

<table>
<thead>
<tr>
<th>Watch Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

☐ Choose the **Watch | Add Field** menu.

☐ Choose **Transaction Code** in the **Fields in WA** drop-down list.

☐ Click the **OK** button.

The system adds the Transaction Code variable to the lower pane, and it displays the current value of the Transaction Code variable. In this case, the current value is 900 which uniquely identifies the Location Status business process.

<table>
<thead>
<tr>
<th>Watch Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Transaction Code</td>
</tr>
</tbody>
</table>

☐ Click the **Step Into** button of Visual Debugger **four times**.

At this point in the logic, the system updates the Location field to the value entered by the end user on the Virtual Terminal. The bottom panel of Visual Debugger reflects this change and displays the newly changed field value in red.

<table>
<thead>
<tr>
<th>Watch Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Transaction Code</td>
</tr>
</tbody>
</table>

☐ Click the **Step Into** button of Visual Debugger **thirteen times**.
At this point in the logic, the system updates the Status field to the status of the given location as defined in the location master database table. (In this case, the status is E.) The bottom panel of Visual Debugger reflects this change and displays the newly changed field value in red.

<table>
<thead>
<tr>
<th>Watch Window</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Transaction Code</td>
</tr>
</tbody>
</table>

You are done utilizing the Watch feature, so now you will exit the tool.

- Choose the **File | Detach Debugger** menu in Visual Debugger.

The system closes the Visual Debugger. The Virtual Terminal advances to next prompt. And Virtual Terminal returns to normal operation.

- In Virtual Terminal, navigate back to the **menu**.
Activity 4: Add a Breakpoint

**Background:** A breakpoint allows you to stop the execution of a process object at a specific line number. You set a breakpoint near the place where you suspect the error. Then you run the business process from the terminal. The terminal will run normally, but slowly, until it reaches the breakpoint. When it reaches the breakpoint, the Visual Debugger captures control of the business process. Then you can step through the process object lines one by one, just like you did in a previous activity.

**Scenario:** This scenario builds upon the one from the previous activity. The HighJump support team has reviewed the watch variable information you read to them, but they have not found the root cause of the problem. They suspect the problem is inside a process object called “Log – Location Status”. They ask you to set a breakpoint in this process object in order to assist with the troubleshooting process.

**Procedures:** The procedures below demonstrate how to add a breakpoint to Visual Debugger. However, they do not reproduce the problem with the incorrect data going to the host system.

- Open the Virtual Terminal, if necessary.
- Navigate to the User ID prompt of the Logon process.
- Open the Visual Debugger.
- Attach the Visual Debugger to the Virtual Terminal.
- Enter a user at the User ID prompt of the Virtual Terminal

The system displays several lines of the current process object.

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Pass</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>Calculate : RF Mask On := &quot;FALSE&quot;</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>19</td>
<td>Call : Enhanced RF Constructor</td>
<td>21</td>
<td>47</td>
</tr>
<tr>
<td>21</td>
<td>Dialog : User ID [Title]</td>
<td>27</td>
<td>22</td>
</tr>
<tr>
<td>22</td>
<td>Compare : <a href="">T5:Version</a>?</td>
<td>25</td>
<td>23</td>
</tr>
<tr>
<td>23</td>
<td>Compare : <a href="">T2:Full</a>?</td>
<td>24</td>
<td>51</td>
</tr>
<tr>
<td>24</td>
<td>Call : List Active Locales</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>Call : Display UI Version</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>Call : DB’Connect, Validate User ID</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>Compare : TWO Excerpt := &quot;WITH BACK&quot;</td>
<td>19</td>
<td>30</td>
</tr>
</tbody>
</table>

- Choose the Debug | New Breakpoint menu in Visual Debugger.

The system opens the Add a Breakpoint window.
Choose **Log – Location Status** in the **Process Object** drop-down list.

Click the **Get** button.

The system displays the definition of the Log – Location Status process object.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2:List</td>
<td>LOG – Clear List</td>
</tr>
<tr>
<td>4:Call</td>
<td>Calculate LOG Elapsed Time</td>
</tr>
<tr>
<td>5:Calculate</td>
<td>Stored Att ID = &quot;&quot;</td>
</tr>
<tr>
<td>6:Call</td>
<td>Set Generic Attribute Values</td>
</tr>
<tr>
<td>7:Calculate</td>
<td>LOG Fill – Generic Attributes</td>
</tr>
</tbody>
</table>

Double click in the **white space** to the **left of the number 9**.

The system places a stop sign icon to the left of the selected line. The stop sign indicates the breakpoint.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2:List</td>
<td>LOG – Clear List</td>
</tr>
<tr>
<td>4:Call</td>
<td>Calculate LOG Elapsed Time</td>
</tr>
<tr>
<td>5:Calculate</td>
<td>Stored Att ID = &quot;&quot;</td>
</tr>
<tr>
<td>6:Call</td>
<td>Set Generic Attribute Values</td>
</tr>
<tr>
<td>7:Calculate</td>
<td>LOG Fill – Generic Attributes</td>
</tr>
<tr>
<td>9:Calculate</td>
<td>LOG Fill – Location Status</td>
</tr>
<tr>
<td>10:List</td>
<td>LOG – Append Fields</td>
</tr>
</tbody>
</table>

Click the **Done** button.

Choose the **Debug | Go** menu.

When you click the Go menu, the Virtual Terminal will run normally until it encounters a breakpoint or until you click the Step Into button of the Visual Debugger.
Navigate to the Location Status prompt of the Location Status business process in the Virtual Terminal.

The Virtual Terminal runs normally, but slowly. There is no activity in the top pane of the Visual Debugger. And the Messages tab of Visual Debugger is empty. The system has not encountered the breakpoint yet.

Enter a status at the Location Status dialog on the Virtual Terminal.

The Virtual Terminal does not advance to the next prompt. Additionally, the top pane of the Visual Debugger window displays the definition of the Log – Location Status process object. The yellow arrow, which indicates the next line the system will execute, is standing on the line number 9. Visual Debugger has finally reached the breakpoint. You can still see the icon.

At this point Visual Debugger has control of the process, and it is waiting for you to step through it line by line. It is in the same state it was during the first activity. You can use the Step Into button to walk through the lines one by one.

Click the Step Into button several times.

Visual Debugger will maintain the definition of this breakpoint even after you stop and start the tool again, unless you tell it otherwise. You can remove all breakpoint definitions through a menu option. (You can disable an individual one by finding the “stop sign” icon you created earlier and double clicking on it.) In this example you will disable all of the breakpoints.

Choose the Debug | Clear All Breakpoints menu.

You are done utilizing the Breakpoint feature, so now you will exit the tool.

Choose the File | Detach Debugger menu in Visual Debugger.

The system closes the Visual Debugger. The Virtual Terminal advances to next prompt. And Virtual Terminal returns to normal operation.
In Virtual Terminal, navigate back to the menu.
Activity 5: Use the Message Watch

Background: In a previous activity, you displayed the line-by-line output in the messages tab of the lower pane of Visual Debugger. You did this by using the Step Into button. However, if you are only interested in viewing the line-by-line output, then the Step Into button is not very efficient. The Step Into button only gives you one line of output per mouse click. The Message Watch solves this problem. The Message Watch feature allows you to view a multitude of line-by-line outputs with only a couple mouse clicks.

Scenario: This scenario builds upon the one from the previous activity. The HighJump support team has analyzed the information you read to them related to the breakpoint, but they have not found the root cause of the problem. They ask you to run several iterations of the Location Status business process, and then send them the line-by-line output for each iteration.

Procedures: The procedures below demonstrate how to enable the Message Watch feature. However, they do not reproduce the problem with the incorrect data going to the host system.

- Open the Virtual Terminal, if necessary.
- Navigate to the Location prompt of the Location Status business process.
- Open the Visual Debugger.
- Attach the Visual Debugger to the Virtual Terminal.
- Enter a location at the Location prompt of the Virtual Terminal.

The system displays several lines of the current process object.

![Screenshot of Visual Debugger with Message Watch enabled]

- Click the Messages tab along the bottom of the Visual Debugger.

The system displays the current line-by-line output.
Choose the **Tools | Message Watch** menu.

There are no immediate visual cues that you have enabled Message Watch mode. However, if you return to the Tools menu, you will see a checkmark next to the Message Watch menu. This indicates that Message Watch mode has been enabled.

Choose the **Debug | Go** menu.

As the system executes a line of code it outputs one or more lines of data to the Messages tab. It will do this continuously until you manually disable the Message Watch mode or until you shut down Visual Debugger. The performance of the Virtual Terminal is significantly slower when the Message Watch is enabled. However it does advance to the next prompt after you enter a piece of data.

Perform a couple iterations of the **Location Status** business process on the Virtual Terminal.

The system continues to scroll the line-by-line output to the messages tab.
After you have captured the desired line-by-line output, you can copy the data out of the messages tab and paste it into another document.

☐ Create a new text file on the desktop.

☐ Open the text file with Notepad.

☐ Highlight multiple rows in the Message tab of Visual Debugger with a click and drag methodology.

☐ Press the Control C combination of keys on the keyboard.

☐ Choose the Edit | Paste menu in the Notepad application

The system pastes the text from the Visual Debugger into the Notepad application.

☐ Choose the File | Save menu in the Notepad application

☐ Choose the File | Exit menu in the Notepad application

At this point you could send the text file to the HighJump support team for further analysis. You are done utilizing the Message Watch feature, so now you will exit the tool.

☐ Choose the File | Detach Debugger menu in Visual Debugger.

The system closes the Visual Debugger. The Virtual Terminal advances to next prompt. And Virtual Terminal returns to normal operation.

☐ In Virtual Terminal, navigate back to the menu.
Activity 6: Use the Record Feature

Background: The Record feature of Visual Debugger operates similarly to the Message Watch feature. It maintains the continuous stream of line-by-line output while the terminal runs in a normal mode. However, the Record feature does not direct the line-by-line output to the Messages tab. Instead, it writes the line-by-line output directly to a text file. This simplifies the process of extracting the line-by-line output from Visual Debugger.

Scenario: This scenario is identical to the one from the previous activity. However, you know that you want to capture the line-by-line output and send it to the HighJump support team in a Notepad document. Since the Record feature simplifies the extraction process, you opt to use the Record features instead of the Message Watch feature.

Procedures: The procedures below demonstrate how to enable the Record feature. However, they do not reproduce the problem with the incorrect data going to the host system.

☐ Open the Virtual Terminal, if necessary.

☐ Navigate to the Location prompt of the Location Status business process.

☐ Open the Visual Debugger.

☐ Attach the Visual Debugger to the Virtual Terminal.

☐ Enter a location at the Location prompt of the Virtual Terminal.

The system displays several lines of the current process object.

```
22:Compare   : SCR HELP2 = "r"
30:Dialog    : PRINT-1:OPT-3:LINE-N
31:Dialog    : PRINT-1:OPT-2:LINE-S
33:Compare   : SCR HELP2 = "r"
34:Dialog    : PRINT-2:OPT-2:LINE-N
35:Dialog    : PRINT-2:OPT-1:LINE-S
37:Return    : PASS
38:Return    : FAIL
```

☐ Click the Messages tab along the bottom of the Visual Debugger.
The system displays the current line-by-line output.

Choose the **Tools | Record** menu.

The system displays a message indicating that breakpoints will not be honored while the Visual Debugger is in Record mode.

Click the **Yes** button.

There are no immediate visual cues that you have enabled the Record feature. However, if you return to the Tools menu, you will see a checkmark next to the Record menu. This indicates that the Record feature has been enabled.

Click the **leftmost button** on the toolbar.

This button is called the Go button, and it is the equivalent of choosing the Debug | Go menu.
Perform a couple iterations of the **Location Status** business process on the Virtual Terminal.

The Virtual Terminal runs slowly, but it advances to the next prompt every time you enter a piece of data. However, Visual Debugger does not write any data to the messages tab. Behind the scenes Visual Debugger is writing the line-by-line output to a text file.

Choose the **File | Detach Debugger** menu.

Enter a value at the **current prompt** in the **Virtual Terminal**, if necessary, in order to close the Visual Debugger.

In Virtual Terminal, navigate back to the **menu**.

At this point, you have captured all of the necessary line-by-line output. Now, you can locate the text file and send it to the HighJump support team.

Open **Windows Explorer**.

Navigate to the **D:\ProgramData\HighJump Software\CONTROL\CONTROL1** folder.

The name of the desired file begins with **R**, and it contains the ID of the terminal and has an **LOG** extension. In this case the file name is **R_RAD01_record**.

Double-click the **R_RAD01_record** text file.
The system displays the line-by-line out from the Visual Debugger in the Notepad application.

![Visual Debugger Output]

☐ Choose the **File | Exit** menu in the **Notepad** application.

At this point, you could zip the file if necessary, and then send it to the HighJump support team for further analysis.
Activity 7: View the Call Stack

**Background:** Every business process which runs on an RF terminal is configured in Advantage Architect using a series of process objects. These process objects are built in layers in which the top-level process object calls a 2nd-level process object which calls a 3rd-level process object. This pattern can continue for several layers depending on the complexity of the requirements.

When debugging an issue with a business process, it is often beneficial to know the name of the current process object which is being executed as well as the names of all the other process objects which are layered above it. The list of the layered process objects is called a call stack. Visual Debugger, as well as the AWESM tool, have features that display the process object call stack.

**Procedures:** Perform the following procedures to view the process object call stack in Visual Debugger and in the AWESM tool.

- Open the Virtual Terminal, if necessary.
- Navigate to the Location prompt of the Location Status business process.
- Open Visual Debugger.
- Attach Visual Debugger to the Virtual Terminal.
- Enter M103 at the Location prompt of the Virtual Terminal.

The system displays several lines of the current process object in Visual Debugger.

```
22:Compare : SCR HELP2 = "n"
30:Dialog : PRINT-1:OPT-3:LINE-N
31:Dialog : PRINT-1:OPT-2:LINE-S
33:Compare : SCR HELP2 = "n"
34:Dialog : PRINT-2:OPT-2:LINE-N
35:Dialog : PRINT-2:OPT-1:LINE-S
37:Return : PASS
38:Return : FAIL
```

- Click the Call Stack tab along the bottom of the Visual Debugger.

Visual Debugger displays the process object call stack. The name of the current process object is positioned at the top of the list. The first process object is positioned at the bottom of the list. In this example, the current process object is 10 layers deep.
If the only information you want to capture is the process object call stack, then the above methodology is a bit lengthy. The methodology requires you to open Visual Debugger; attach it to a terminal; and then respond to the screen on the terminal. The methodology works well if you are already in Visual Debugger for other purposes. However, the most efficient way to capture the process object call stack is to use the AWESM tool. Follow the procedures below to view the process object call stack in the AWESM tool.

- Open the Advantage Workflow Engine Service Manager, if necessary.
- Click the WA solution environment, in order to highlight it.
- Click the Sessions button.

The system opens another window which displays all of the sessions for the WA solution environment.

- Click the RADIO_01 device, in order to highlight it.
- Click the Status button near the lower left corner of the window.

The system opens a large window that displays several status-related properties of RADIO_01. The system displays the process object call stack near the bottom of the window. Note that the call stack in the AWESM tool is the same information as the call stack in Visual Debugger.
Admin Lab Exercise 12:
Visual Debugger

Sometimes when you are debugging a business process, it may be beneficial to copy the call stack information and send it to a third party for their analysis. The AWESM tool allows you to copy the call stack, but Visual Debugger does not. Follow the procedures below to copy the call stack and paste it into a Notepad file.

1. Highlight the entire contents of the Process Object Call Stack box in the AWESM tool using the mouse.
2. Right-click the selected text.
3. Choose the Copy menu option.
4. Open the Notepad application.
5. Choose the Edit | Paste menu option.

The system pastes the call stack information into Notepad.

1. Save the Notepad file to the desktop.
2. Close the Notepad application.
3. Click the “X” in the upper right corner of the Status for [RADIO_01] window to close it.
4. Click the “X” in the upper right corner of the Sessions – WA window to close it.
5. Choose the File | Detach Debugger menu.
6. Enter a value at the current prompt in the Virtual Terminal, if necessary, in order to close the Visual Debugger.
7. In Virtual Terminal, navigate back to the menu.
Activity 8: View a List

In some business processes, the system reads several records from the database, and then stores them in the memory of the application server. In these business processes, the system usually iterates through each record and then performs some work on each one. The collection of records which is stored in the memory of the application server is called a list. When debugging an issue with a business process, it may be beneficial to view the contents of a list.

The procedures below utilize a business process, called View by Item, which allows you to view all inventory locations containing a given item number. The business process uses a list to present the locations to the end user. Follow the procedures below to view the contents of the list used in the View by Item business process.

☐ Open the Virtual Terminal, if necessary.

☐ Navigate to the Main Menu screen.

☐ Press the **F8 key** in order to page down to the next set of menu options.

☐ Enter **7 (Inv Control)** at the **Option** prompt.

☐ Enter **3 (Inventory Status)** at the **Option** prompt.

☐ Enter **2 (View by Item)** at the **Option** prompt.

The system advances to the first prompt of the View by Item business process.

☐ Enter **NNFG00000** at the **ITEM ID** prompt.

Behind the scenes, the system retrieves several inventory-related records from the database and stores them in a list on the application server. The system then displays several inventory attributes of the first location containing the given item number.
Admin Lab Exercise 12:
Visual Debugger

- Press the **F8:Next** key **three times** to navigate through the first three records.
- Press the **F4:Prev** key **three times** to navigate back to the first record.
- Open **Visual Debugger**.
- Attach Visual Debugger to the Virtual Terminal
- Press the **F8:Next** key on the Virtual Terminal to advance to the next location containing the item.

Visual Debugger grabs control of the business process and displays several lines of the current process object in Visual Debugger.

Now that Visual Debugger is attached to the terminal, you will view the contents of the list containing the inventory-related data.

- Choose the **Watch | Add List** menu option.

The system displays the Add Watch List window and requires you to choose the name of the list. There are several lists displayed in the drop-down control. The name of the list is determined by the developer at the point of configuration. In this example the inventory-related information is stored in a list called View Inventory List.
Choose **View Inventory List** in the **Lists in WA** drop-down list.

Click the **OK** button.

The system displays the data in the list using a grid, and it displays several other attributes of the list immediately above the grid. Note that the data in row number 1 of the list is the same data on the Virtual Terminal. By default, the system displays the first 10 rows of the list. But in this example, there are a total of 28 rows in the list.

If you want to see more than 10 rows in the list, you can change the limit and then refresh the grid.

Enter **50** in the **Limit** edit box.

Click the **Refresh** button.

The system refreshes the data in the grid based upon the new settings. The system now displays a vertical scroll bar in the grid and allows you to view up to the first 50 rows in the list.
Depending on your Advantage Architect skills, it may be beneficial to copy the list information out of the grid and send it to a third party for their analysis.

☐ Press and hold the **Shift** key.

☐ Click anywhere in **Row 6** to highlight the first 6 rows in the grid.

☐ Press the **Control-C** keys to copy the selected text into the Windows clipboard.

☐ Open the **Notepad** application.

☐ Choose the **Edit | Paste** menu option.

The system pastes the data from the list into Notepad.

☐ Save the **Notepad** file to the desktop.

☐ Close the **Notepad** application.

☐ Choose the **File | Detach Debugger** menu option in **Visual Debugger** to close it.

At this point you could send the text file to a colleague or a support team member for additional analysis.
Admin Lab Exercise 13:
Specific Scenarios

Introduction

This exercise consists of the following activities:

1. Understand the “Log Off Radio” error message
2. Understand the “In Use By” error message
3. Shadow a Telnet Session

Background Information

The activities below reproduce various scenarios that you may encounter as an administrator. Some of the scenarios can be resolved with simple corrections. Others require the assistance of the HighJump support team. The emphasis in these activities is on understanding what produces the given symptoms.

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the Troubleshooting Scenarios menu. (The link is located on the right side panel of the VLab interface.)
Activity 1 : Understand the “Log Off Radio” Error Message

**Background:** The HighJump system prevents a material handler (employee) from logging into multiple terminals at the same time. If a material handler attempts to log into a terminal while he is concurrently logged into a different terminal, the system gives the error message “LOG OFF RADIO XX” and it does not allow him to log in a second time.

This specific error message is only relevant to the WA application. The DCA application uses a different set of business processes, and it will not produce this symptom.

**Scenario:** Amy is logged into an RF terminal and is in the middle of the order picking process. The system directs her to pick product from a location in the back of the warehouse. She hops on a forklift and begins driving towards the location. On the way to the location, she accidentally drops the RF terminal and runs over it with her forklift. The forklift destroys the terminal to a point beyond recognition. She reports this situation to the shift supervisor, who reprimands her and then gives her another terminal to continue her picking tasks.

**Procedures:** The procedures below mimic the scenario above and demonstrate how the system behaves under those conditions.

- [ ] Start the engine for the **WA solution environment**, if necessary.
- [ ] Open a **Virtual Terminal**, if necessary.
- [ ] Log into the Virtual Terminal as **AMY | AMY | FAAMY** to reach the main menu.
- [ ] Enter **4** (Order Pick) at the **OPTION** prompt.
- [ ] Enter **1** (Order Picking) at the **OPTION** prompt.

The system advances to the first prompt of the Order Picking business process.

![Order Picking](image)

Enter PICK AREA - Order Pick.

PICK AREA
In the scenario, Amy drops her RF terminal and runs over it with a forklift. In order to simulate that scenario, you will exit the Virtual Terminal.

- Right-click on the face of the Virtual Terminal.
- Choose the Exit menu.

In the scenario, Amy’s supervisor gives her another RF terminal to continue with the picking tasks. In order to simulate that scenario, you will open a Web Terminal session and attempt to login with Amy’s profile.

- Open a Web Terminal, if necessary. (See Exercise 5: Adding Virtual Terminals and Web Terminals for additional instructions.)
- Enter AMY at the USER ID prompt.

The system displays an error message similar to “LOG OFF RADIO_01.”

```
HighJump Software
Warehouse Advantage
Version 12.14
LOG OFF RADIO_01
USER ID

F5:Version
```

The system does not allow Amy to log in to the Web Terminal because she is already logged in to another terminal. One way of solving this type of issue is to locate the first terminal; log Amy out of it; and then log into the second terminal with Amy’s profile. However, in the painted scenario, the forklift destroyed the terminal, and it is impossible to log Amy out of it. In this specific case, one way to cleanly resolve the issue is to shadow Amy’s terminal. The concept of shadowing a terminal is covered a later activity.

**Follow Up:** At this point, the activity is finished. Perform the following procedures to restore the terminals back to a working state.
- Open the **Virtual Terminal**.
- Press the **F1 key** multiple times to navigate back to the **USER ID** prompt.
Activity 2: Understand the “In Use By” Error Message

Background: The HighJump system prevents multiple material handlers (employees) from utilizing the same piece of equipment (forklift) at the same time. If a material handler attempts to utilize a forklift that is already in use by a different material handler, the system gives an error message “IN USE BY XX” and it does not allow the second material handler to use that forklift.

This specific error message is only relevant in the WA application. The DCA application uses a different set of business processes, and it will not produce this symptom.

Scenario: This scenario is similar to the one from the previous activity. However, it focuses on the forklift instead of the terminal. Amy is logged into an RF terminal and is in the middle of the order picking process. The system directs her to pick product from a location in the back of the warehouse. She hops on a forklift and begins driving towards the location. On the way to the location, she accidentally drops the RF terminal and runs over it with her forklift. The forklift destroys the terminal to a point beyond recognition. She reports this situation to the shift supervisor, who reprimands her and then gives her another terminal to continue her picking tasks. In light of the day’s events, Amy needs to take a break before continuing her picking tasks. She parks the forklift in the forklift bay, and then heads to the lunchroom for a much-needed break. While she is in the lunchroom, Pat begins his shift. He grabs an RF terminal, hops on Amy’s unused forklift in the bay, and attempts to login to the terminal and the forklift.

Procedures: The procedures below mimic the scenario above and demonstrate how the system behaves under those conditions.

- Start the engine for the WA solution environment, if necessary.
- Open a Virtual Terminal, if necessary.
- Log into the Virtual Terminal as AMY | AMY | FAAMY to reach the main menu.
- Enter 4 (Order Pick) at the OPTION prompt.
- Enter 1 (Order Picking) at the OPTION prompt.

The system advances to the first prompt of the Order Picking business process.
In the scenario, Amy drops her RF terminal and runs it over with a forklift. In order to simulate that scenario, you will exit the Virtual Terminal.

☐ Right-click on the face of the Virtual Terminal.

☐ Choose the Exit menu.

In the scenario, Amy takes a break and Pat attempts to utilize the unused forklift. In order to simulate that scenario, you will open a Web Terminal session and login with Pat’s profile. At the EQUIPMENT \ ZONE prompt, you will attempt to enter the forklift that Amy was using (FAAMY).

☐ Open a Web Terminal, if necessary.

☐ Enter PAT at the USER ID prompt.

☐ Enter PAT at the PASSWORD prompt.

☐ Enter FAAMY at the EQUIPMENT \ ZONE prompt.

The system displays an error message “IN USE BY AMY”.

Order Picking

Enter PICK AREA - Order Pick.

PICK AREA

-
The system does not allow Pat to use the FAAMY forklift because it is still associated with Amy’s profile. One way of solving this type of issue is to locate the first terminal; log Amy out of it; and then log into the second terminal with Pat’s profile and the FAAMY forklift. However, in the painted scenario, the forklift destroyed the terminal, and it is impossible to log Amy out of it. In this specific case, one way to cleanly resolve the issue is to shadow Amy’s terminal. The concept of shadowing a terminal is covered in a later activity.

**Follow Up:** At this point, the activity is finished. Perform the following procedures to restore the terminals back to a working state.

- Open the Virtual Terminal.
- Press the **F1 key** multiple times to navigate back to the **USER ID** prompt.
- In the Web Terminal, press the **F1 key** multiple times to navigate back to the **USER ID** prompt.
Activity 3: Shadow a Telnet Session

In the previous activities you examined a couple scenarios in which the RF terminal was out-of-commission, but still had a user who was logged into it. In the previous activities, you saw some error messages that resulted from these situations. However, you did not learn how to recover from them. This activity provides some direction on how to cleanly recover from these situations.

This training class does not provide you with a physical RF terminal. Instead of giving exact procedures on how to cleanly recover from an out-of-commission RF terminal (which you would not be able to test), this activity gives a high-level overview of the process.

The HighJump application server and the RF terminals communicate with one another via a communication protocol called telnet. HighJump uses a solution from Georgia SoftWorks to provide the telnet communication software and the associated telnet administrative tools. If you need to cleanly recover from one of the above situations, you can use the Georgia SoftWorks tools.

One of the tools provided by Georgia SoftWorks is called the Session Administrator. And within this tool, there is a feature called Shadowing. The Shadowing feature allows you to connect to and interact with a telnet session. In the example of the out-of-commission RF terminal, you can launch the Shadowing feature and then point to the telnet session associated with the RF terminal. When you do this, Georgia SoftWorks will display a screen that is virtually identical to what the operator would see on the RF terminal if the RF terminal was still functioning. For example, if a forklift operator ran over the RF terminal while it was on the LOCATION screen of the Order Picking business process, then the Georgia SoftWorks administrator will see the LOCATION screen of the Order Picking business process when he Shadows the telnet session. Additionally, the administrator can interact with the business process. In this example, the administrator can manually finish the Order Picking process (through the Georgia SoftWorks Shadowing feature) by responding to the Order Picking screens using the server keyboard. When he finishes the Order Picking process, he can then use the standard F1 key to navigate back to the USER ID screen.

With the Shadowing feature, the administrator can cleanly finish the Order Picking process without any risky manual updates to the database tables. The Shadowing feature provides a proven method for cleanly completing a business process on an out-of-commission RF terminal.

For additional details regarding the Shadowing feature of the Georgia SoftWorks Session Administrator, please visit the website below.

Page Editor Lab Exercise 14: Page Editor Applications

Introduction

This exercise consists of the following activities:

1. Backup the Training Page Editor Application
2. Export a Page Editor Application
3. Import a Page Editor Application
4. Publish a Page Editor Application
5. Restore the Training Page Editor Application
6. Create a New Page Editor Application
7. Working with Page Locking

Background Information

If you make a series of changes to a Page Editor application in a development environment, then eventually you will need to move those changes to the production environment. When you move the changes to the new environment, you will need to perform an export, an import, and a publish as part of the process. Basically, anytime you move an application from one machine to another you will need to run an export, an import, and a publish as you backup and replace the files.

Before performing this exercise, we recommend you use the Pre-Work: Advantage System Administration link and review the related video demonstrations under the PE Apps – Export and Import menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Backup the Training Page Editor Applications

**Background:** All exercises, up until this point, have had you modify the Advantage Dashboard and Warehouse Advantage Page Editor applications. However, in the upcoming import/export and release/promotion exercises you will overwrite these applications. A utility script has been provided that will back up your work before you overwrite these applications. Upon completion of these exercises, you will later be directed to run a similar utility script to restore these applications. Collectively, these activities will allow you to continue where you left off prior to import/export and release/promotion exercises.

Follow the procedures below to back up your Advantage Dashboard and Warehouse Advantage Page Editor applications.

- Choose the **File | Exit** menu option to close Page Editor for **WAWarehouseAdvantageConfig** (if currently opened).
- Choose the **File | Exit** menu option to close Page Editor for **WAAvantageDashboardConfig** (if currently opened).
- Choose the **File | Exit** menu option to close Page Editor for **WebUserManagementConfig** (if currently opened).
- Open the **Training** folder on the **desktop**.
- Navigate to the **Page Editor | Utility** folder.
- Right-click the **pe_class_utility.bat** file.
- Choose the **Run as Administrator** menu option.

The system opens a utility that was built specifically for this training class.

- Enter **C (Back Up Exercise Configs)** at the **Selection** prompt.
The system displays a dialog asking if you want to continue with the changes.

☐ Enter Y at the Proceed prompt.

The system shows the results.

☐ Press the Enter key.

At this point, you have set up all of the necessary components of the custom-built Page Editor application. The system now returns to the main menu of the utility.

☐ Enter 0 at the Selection prompt.

The system exits the utility. At this point all of the components are in place.
Activity 2: Export a Page Editor Application

Background: Because of the backend architecture, Page Editor does not have an export feature built into the tool. Whenever you need to export a Page Editor application, you locate a single file on the hard drive; copy the file; and then paste that file into a different folder. This simple copy/paste mechanism serves as the export.

Scenario: Your development team has made a series of changes to the Advantage Dashboard Page Editor application in the development environment. It is your responsibility to export the Advantage Dashboard application from the development environment and place the Page Editor configuration file in a folder on the network which will be imported into the test environment at a later date.

Procedures: The procedures below demonstrate how to export the Advantage Dashboard application.

First, you will create a folder structure on the desktop which will hold the exported application. In the real-world, this folder would likely be on a network share. Since this scenario is in the context of promoting an application from one environment to another, you will create a folder name that contains a sequential release number. The folder structure you create will look like similar to the following screenshot.

- Right-click anywhere on the Desktop.
- Choose the New | Folder menu option.
- Enter My Releases in the edit box for the name of the folder.
- Double click the My Releases folder you just created in order to open it.
- Right click anywhere in the panel on the right.
- Choose the New | Folder menu option.
- Enter Release 01 in the edit box for the name of the folder.

The folder structure should look like the following snapshot.
Double click the **Release 01** folder you just created in order to open it.

Next you will navigate to the HighJump folder which holds the page definitions for the Advantage Dashboard Page Editor application.

- Open **Windows Explorer** in another browser window.
- Navigate to the `D:\ProgramData\HighJump Software\WebConfigs` folder.

This folder contains several files with WDB extensions. Each file holds the page definitions for a different web application.

Shortly, you will work with the web application called **WAAdvantageDashboardConfig.wdb**. If this folder contains a **WAAdvantageDashboardConfig.LDB** file then somebody has the **WAAdvantageDashboardConfig** application open in the Page Editor development tool. Best practice says that you should not operate on the WDB file if the corresponding LDB file exists.

- If **WAAdvantageDashboardConfig.idb** exists in this folder, then close the **Advantage Dashboard** instance of **Page Editor**.

- Copy the **WAAdvantageDashboardConfig.wdb** file.

- Paste the **WAAdvantageDashboardConfig.wdb** file into the **Release 01** folder you created earlier.
The folder should now look like the following screenshot.

![Folder screenshot](image)

Additionally, it is good practice to include a text file in the release folder which indicates the exact changes that were made to the application. This text file is called the release notes. In this example, you will create a very simple set of release notes.

- Open Notepad
- Add the following text to the Notepad file.

```
----------------------
Release 01
----------------------
1. Advantage Dashboard: includes all training exercises from class
```

The Notepad file should look similar to the following screenshot.

![Notepad screenshot](image)

- Save the Notepad file to the Release 01 folder you created earlier with a filename of `release_notes_for_release_01`.

The Release 01 folder should look like the following screenshot.

![Folder screenshot](image)
☐ Close the **Notepad** application

If there were additional applications, scripts, reports, or labels that needed to be promoted from the development environment to the test environment, you could include those files in the Release 01 folder and then update the release notes accordingly. In this example, Advantage Dashboard is the only application needs to be promoted, and the release folder now contains everything necessary to promote the changes to the next environment. The export process is now complete.
Activity 3: Import a Page Editor Application

Background: In the previous activity you learned that exporting a Page Editor application consists of manually copying a file from the hard drive and manually pasting it into a different folder. An import works in the opposite direction. An import of a Page Editor application is very similar. To import a Page Editor application you must also manually copy and paste a file from one folder into another.

When you import a Page Editor application you will not automatically see the changes reflected in the HighJump One Platform UI. In order to see the changes you must also go through a publishing process.

Scenario: The Warehouse Advantage HighJump One Platform UI menu contains three menu options which your company does not use: Cross Dock Profiles, Customers, and Dynamic Item Attributes. You would like to remove those options from the menu structure.

The diagram below shows the current state of the Warehouse Advantage menu. It also shows how you want the new menu to look after the change has been made.

Your company does not do any Page Editor development, so you contract with the HighJump Professional Services team to develop the menu changes. In reality, this is a very simple application change. You would probably not request this change from HighJump unless it was bundled with some other more significant changes. It is included here simply to demonstrate the concept.

In order to start the process, you make a copy of the Warehouse Advantage Page Editor config file, and you send it to the HighJump team. The HighJump team changes the menu, and then they send you a modified Warehouse Advantage Page Editor config file. Your task is to import the configuration file into Page Editor on the development server, and then push the changes out to the development the HighJump One Platform UI. This activity demonstrates the import component. The next activity demonstrates the “push” component.
**Current State:** In this exercise you will view the application in its current state, so that you can determine whether or not the import and publish of the new application were successful. The chart below outlines the test cases to consider, as well as the expected outcome. Execute each test case to familiarize yourself with the current state of the business logic. You will revisit these same test cases after you have performed the import and the publish.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Current State of the Business Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the HighJump One Platform UI, expand the **Supply Chain Advantage</td>
<td>Warehouse Advantage** menu.</td>
</tr>
</tbody>
</table>

![Diagram](diagram.png)

**Procedures:** The procedures below demonstrate how to import a new Warehouse Advantage Page Editor application into the Page Editor tool. The imported application removes the three unused menu options from the primary menu.

First you will copy the configuration file provided by the HighJump PSG team and paste it into the appropriate directory in the development environment.

- Open the folder on the **desktop**.
- Navigate to the **Releases \ Release 04 ...** folder.
The system shows that there are two files in this folder.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>release_notes_for_release_04</td>
<td>6/23/2015 10:03 AM</td>
<td>Text Document</td>
<td>1 KB</td>
</tr>
<tr>
<td>WAWarehouseAdvantageConfig_release_04.wdb</td>
<td>2/15/2016 8:55 AM</td>
<td>WDB File</td>
<td>4,000 KB</td>
</tr>
</tbody>
</table>

☐ Copy the **WAWarehouseAdvantageConfig_release_04.wdb** file.

☐ Navigate to the `D:\ProgramData\HighJump Software\Web Configs` folder.

☐ Paste the **WAWarehouseAdvantageConfig_release_04.wdb** file into this folder.

If this folder contains a **WAWarehouseAdvantageConfig.LDB** file then somebody has the **WAWarehouseAdvantageConfig** application open in the Page Editor development tool. Best practice says that you should not operate on the WDB file if the corresponding LDB file exists.

☐ If **WAWarehouseAdvantageConfig.ldb** exists in this folder, then close the **Warehouse Advantage** instance of **Page Editor**.

The folder should look like the following snapshot.

In order for the solution to work correctly, the new configuration file must be named **WAWarehouseAdvantageConfig.wdb**. This means that you have to get rid of, move, or rename the old configuration file and then rename the new one. As a matter of best practice you should maintain a backup of the original file. The exercise uses a very simple backup strategy. You and your company will need to determine how you maintain backups of your Page Editor configuration files.

☐ Rename **WAWarehouseAdvantageConfig.wdb** to **WAWarehouseAdvantageConfig_original.wdb**
Rename `WAWarehouseAdvantageConfig_release_04.wdb` to `WAWarehouseAdvantageConfig.wdb`.

The folder should look like the following snapshot.

![Folder snapshot]

At this point, the new Warehouse Advantage Page Editor application has been successfully “imported” into Page Editor. Page Editor now has visibility to the modified menu structure. However, in order to see the changes in the HighJump One Platform UI, you will need to publish the application. The next activity demonstrates how to publish the application.
Activity 4: Publish a Page Editor Application

Background: When you “import” a Page Editor application the modifications are not automatically reflected in the HighJump One Platform UI. If you want to “push” the modified application out to the runtime users, then you need to manually publish the application.

Scenario: In the previous activity you imported a new Warehouse Advantage Page Editor application in which the Dynamic Item Attributes, Cross Dock Profiles, and Kits menu options were removed from the menu structure. You performed the import in the development environment. Now you want to publish this new application to HighJump One Platform UI so that you can test the changes made by the HighJump PSG team.

Procedures: The procedures below demonstrate how to manually publish the new Warehouse Advantage Page Editor application to the HighJump One Platform UI.

☐ Choose the Start | All Programs | HighJump Software | WebWise Page Editor menu.

The system opens the Connection window for the Page Editor application.

☐ Choose WAWarehouseAdvantageConfig in the DSN drop down list.

The system opens the Page Editor tool.
Expand the **Publish To** drop down box in the tool bar.

The system displays three Publish To options – one for each of the environments.

In this scenario, the HighJump Software PSG team developed the changes, and then sent you the new application. It is your responsibility to port these changes into the Development environment. Consequently, you will choose Development from the Publish To drop down box.

Choose **Development** in the **Publish To** drop down box in the tool bar.

The tool bar should look like the following screenshot.

When you publish the full application to the Development environment, the system takes the entire set of pages in the given Page Editor application and makes them available to the HighJump One Platform UI in the Development environment.

Choose the **File | Publish | Full** menu.

The system displays a confirmation window.

Click the **Yes** button.

When the publishing process is complete, the system displays a message.
Click the **OK** button.

Choose the **File | Exit** menu.

**Validation:** You have successfully imported and published the Warehouse Advantage Page Editor application. Now you will validate that the HighJump PSG team modified the menu structure correctly. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps in this activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. In the HighJump One Platform UI, choose the **Administrator</td>
<td>Logout** menu option that appears near the upper-right corner. Then click the <strong>Logout</strong> button.</td>
</tr>
<tr>
<td></td>
<td>2. Use the <strong>HighJump One Platform UI #1</strong> section of the <strong>login.txt</strong> file in the folder on the desktop to login.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Click the **Supply Chain Advantage</td>
<td>Warehouse Advantage** menu.</td>
</tr>
</tbody>
</table>
Activity 5: Restore the Training Page Editor Application

Scenario: You have just completed all of the import/export and release/promotion related activities. Before starting these exercises, recall that you ran a utility script to back up your work with the Advantage Dashboard and Warehouse Advantage Page Editor applications. A similar script has also been provided to restore these applications and allow you pick up where you left off.

Follow the procedures below to restore your Advantage Dashboard and Warehouse Advantage Page Editor applications.

- Choose the File | Exit menu option to close Page Editor for WAWarehouseAdvantageConfig (if currently opened).
- Choose the File | Exit menu option to close Page Editor for WAAAdvantageDashboardConfig (if currently opened).
- Open the Training folder on the desktop.
- Navigate to the Page Editor \ Utility folder.
- Right-click the pe_class_utility.bat file.
- Choose the Run as Administrator menu option.

The system opens a utility that was built specifically for this training class.

- Enter D (Restore Exercise Configs) at the Selection prompt.

The system displays a dialog asking if you want to continue with the changes.
Enter Y at the Proceed prompt.

The system shows the results.

Press the Enter key.

At this point, you have set up all of the necessary components of the custom-built Page Editor application. The system now returns to the main menu of the utility.

Enter 0 (Quit) at the Selection prompt.
The system exits the utility. At this point all of the original components are in place. Next you will publish the Page Editor application, using the same process from above.

- Open the **WAWarehouseAdvantageConfig** application in **Page Editor**, if necessary.
- Choose **Development** in the **Publish To** drop down box in the tool bar.
- Choose the **File | Publish | Full** menu.
- Click the **Yes** button.
- Click the **OK** button.
- Choose the **File | Exit** menu.

At this point you have imported the original Page Editor Warehouse Advantage application, and you have published it to the HighJump One Platform UI. If everything was successful, you should see the original menu in runtime. Because this modification has an effect on the menu panel, you must logout and login in order to see the changes.

- In the HighJump One Platform UI, **logout** of the application.
- Use the **HighJump One Platform UI #1** section of the **login.txt** file in the **folder on the desktop** to log in.
- Click the **Supply Chain Advantage | Warehouse Advantage** menu.

The system displays the original menu which contains links for Cross-dock Profiles, Customers, and Dynamic Item Attributes.
In this exercise you exported a Page Editor application as part of a promotion process. Then you imported and published another Page Editor application which was also part of a promotion process. Additionally, you backed up the original Page Editor applications and restored them at end. You are now ready to move on to the next exercise.
Admin Lab Exercise 15: Packages

Introduction

This exercise consists of the following activities:

1. Import a Package
2. Compile and Activate a Package
3. Check In a Package
4. Rollback a Package

Background Information

You learned in a previous exercise that Advantage Architect is one of two primary configuration tools delivered with the Advantage system. When you make changes to an application in Advantage Architect you make changes to a specific object. Some of these objects dictate the layout of the data on the RF screen; some of these objects execute formulas; other objects execute queries against the database; still other objects perform a multitude of different tasks.

Also, in a previous exercise you learned how to export an Advantage Architect application. When you exported the application, you exported the entire application. That is, you exported the entire suite of objects for a given application. Every object, regardless of its type, in the Architect application had a corresponding entry in one of the exported XML files.

A package is an exported subset of objects from an Advantage Architect application. This definition will be refined slightly in the Advantage Architect class, but it will suffice for the Administrator class.

Often times a package is created to address a specific application-layer problem (a hotfix). They are also often created to deliver a requested application-layer change order. For example, if you contract with HighJump Software to make a handful of changes to your Bulk Picking process, HighJump may deliver those changes to you via a package instead of a complete application.

One of the benefits of using a package is that the size of the exported file set is significantly smaller than the entire application. This makes the transfer of data a bit easier.
Another benefit is that it may decrease the amount of testing time. Some companies have a policy that states that every aspect of an application that is moved into a production environment must be tested beforehand. That is, if you import and compile and activate an entire Architect application into production, then you need to test the entire application first. Under this company policy you would need to include all aspects of receiving, picking, shipping, and many more in your testing scenarios. That is no small task. If that is the company policy, then the concept of packages simplifies the testing process. Since a package has a much narrower scope than the entire application, then only those processes impacted by the package would need to be tested. This significantly reduces the amount of time it takes to move the changes into the production environment.

Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Packages menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Import a Package

Scenario: You frequently use the Location Status business process in your operation. However, one drawback to the process is that immediately after entering the new status, the system loops back to the top and prompts for a location. It does not give any indication that it changed the status successfully. The warehouse manager calls you on the phone and asks you to change the application so that it gives some type of visual acknowledgment that it successfully changed the status. After further conversation with the manager, you agree that you would like the process to display the old status and the new status before returning to the location dialog. (See the screen shot below)

Since nobody at your company has the skill set to develop in Advantage Architect, you contract with the HighJump Professional Services Team to make the changes. They develop the changes, and then they send you those application changes via a package. It is your responsibility to import, compile, and activate the changes on the development server.

Procedures: The procedures in this activity demonstrate how to import a package into an Advantage Architect application. The next activity completes the process by demonstrating how to compile and activate an application that contains packages. The package used in these two activities does indeed meet the requirements requested in the scenario. However, you will not be able to test the functionality until you have completed the compile and the activation denoted in the next activity.

☐ Start the engine.
☐ Start a Virtual Terminal.
☐ Execute a couple iterations of the Location Status business process to familiarize yourself with it.
☐ Open Advantage Architect and log into it.

Since you will be replacing a portion of the WA application, it is a good idea to make a backup copy of the existing application. Later in this exercise you will learn how to discard the package even after you have imported it. A full application backup is simply good practice.

☐ Export the WA application into a folder of your choice. (See the activity titled Export an Advantage Architect Application for additional information.)
You now have a full application backup in the event that you need to revert backwards. You can continue with the process of importing the package.

- Choose **WA** in the **Application** drop-down list in the panel on the left.

The left panel of Advantage Architect should look like the following graphic.

- Choose the **File | Import | Import Package** menu.

The system opens the Import Package window.

- Click the **ellipsis button** to the right of the **Package** edit box.

The system opens the standard dialog used to select a file.
☐ Click the **Desktop** button on the left panel.

☐ Navigate to the **Training \ Releases \ Release 07**… folder.

☐ Click the **architect_package_location_status_release_07.pkg** file.

☐ Click the **Open** button.

The system returns to the Import Package window and displays the path and filename of the file you selected on the previous screen.

☐ Click the **OK** button.

The system displays a dialog indicating the package will be imported into the WA application.

☐ Click the **Yes** button.

The system imports the package into the WA application, and it displays the progress in the **Output Window**. When the import is complete, the system displays a message in the **Output Window** instead of a new dialog.
Output Window

- Importing Package [Location Status]
- Verifying package...
- Start importing...
- Package [Location Status] is imported successfully.
Activity 2: Compile and Activate a Package

Scenario: This scenario is identical to the scenario in the previous activity.

Procedures: Importing an Architect package does not automatically impact the business process running on the RF terminals. In order to impact the terminals, you must compile and activate the application. The procedures in this activity demonstrate how to compile and activate an Advantage Architect application that contains a package. The procedures are very similar to the compiling and activating when the application does not contain a package, however there are some small differences.

☐ Choose the Run | Compile Application menu in Advantage Architect.

The system displays the same Compile Application dialog that it displayed for the other compiles.

☐ Choose WA in the Application drop-down list.

The system populates the Compilation Folder edit box with a default value. You can change this folder. However, the default value will suffice for most instances.

The primary difference between the compile with a package and a compile without a package is managed through the Production Compile checkbox. In order to include the package contents in the runtime files, you must uncheck the Production Compile checkbox and then check the With Package Objects checkbox.

☐ Select the Development Compile radio button.

☐ Enable the With Package Objects (Location Status) checkbox.

☐ Click the Compile button.

When the compile is complete, the system displays a dialog indicating that the compile was successful.
From this point forward, the remainder of the compile process and the remainder of the activation process are identical to compiling and activating an application without a package.

- Complete the **compile** and **activate** processes. (see previous exercises for additional instructions)

**Test Cases:** The final step in this process is to validate that the change worked as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

**Starting the Location Status Business Process**

- From the **Main Menu** choose **Option 7, Option 3, Option 4**

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter a valid <strong>location</strong> (M101, M102…) at the <strong>Location</strong> prompt of the <strong>Location Status</strong> business process.</td>
<td>System advances to the Status prompt.</td>
</tr>
<tr>
<td></td>
<td>Enter a valid <strong>status</strong> at the <strong>Status</strong> prompt.</td>
<td>System advances to a new dialog that displays the old status and the new status.</td>
</tr>
</tbody>
</table>
Press the **Enter** key at the **Confirmation** prompt.

System returns to the Location prompt.
Activity 3: Check In a Package

**Background:** When you import a package into an application, the package is treated like a separate module. Consequently, you can easily remove the package if you find that it doesn't meet the requirements. The package behaves like a “bolt on” component that can be removed if desired.

However, the “bolt on” nature of a package should only be considered temporary. Once the functionality of the package has been tested and verified, the package should be incorporated into the full application. The process of incorporating a package into the rest of the application is called “checking in.”

The process of checking in a package is not difficult. However, once a package is checked into the application, it is very difficult to reverse. The check in process is better suited for a developer instead of an administrator. As a result, the Administration class does not detail this process. If you are interested in learning more about this topic, you can research the Architect help files, or you can take the Advantage Architect class offered by the HighJump training department.
Activity 4: Rollback a Package

Background: In the previous activity, you learned that an imported package operates like a “bolt on” component or a separate module. If you find that the package does not meet the desired functionality, you can easily “throw away” or remove the package. The process of throwing away a package is called rolling back a package. However, there is one condition. You cannot rollback a package if the package has already been checked in. Once a package has been checked into the application, it takes a significant amount of effort to remove it.

Scenario: In the previous activities you imported, compiled, and activated an application that added a dialog to the Location Status business process. The warehouse manager has reviewed and tested the new functionality on the development server, and he agrees that the delivered functionality meets the requirements that he requested. However, now that he sees the functionality he wants to take a different course of action. Instead of displaying the old and new statuses, he simply wants to display the standard Transaction Complete dialog at the end of the business process. He asks you to remove the new dialog, and then work with the HighJump development team to add the Transaction Complete dialog in its place.

You opted not to make a full application backup before you imported, compiled, and activated the package. The only option you have for removing the new dialog is to rollback the package.

Procedures: The procedures in this activity demonstrate how to rollback a package. The procedures use the package that was imported, compiled, and activated in the previous activities. However, the procedures do not demonstrate how to add the Transaction Complete dialog to the end of the Location Status business process.

☐ Choose the Version Control | Pending Checkins menu in Advantage Architect.

The system displays the Pending Checkins window, and there are no entries displayed in the window.

☐ Click the second button from the right in the Pending Checkins window.

The system displays a list of objects in the Pending Checkins window. Each of the displayed objects belongs to the Location Status Summary Dialog package that you imported.
Right-click the **WA (PKG_Location Status)** node at the top of the tree.

The system highlights the node, and it displays a submenu.

Choose the **Rollback Package** menu.

The system essentially “throws away” the package, and leaves the Pending Checkins window empty. The package is no longer a “bolt on” component of the application. Advantage Architect no longer has visibility to the package.
Even though Advantage Architect has no visibility to the package, the Virtual Terminal and all other terminals will still display the new dialog in the Location Status business process at runtime. Rolling back the package only impacted the definition of the application in Advantage Architect. In order to change the behavior at runtime, you must compile and activate the current application.

You have seen two different paths for compiling and activating an Advantage Architect application. One path accounts for a package, and the other does not. Since Architect no longer has visibility to the package for the Location Status business process, you must choose the compilation approach that does not include the package.

☐ Compile and activate the application. (see previous exercises for additional instructions)

Test Cases: The final step in this process is to validate that the change worked as expected. The chart below outlines the various test cases that must be validated. Each test case defines the scenario that must be created as well as the expected outcome. If a given test case does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

<table>
<thead>
<tr>
<th>Pass / Fail</th>
<th>Test Case</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter a valid location (M101, M102…) at the Location prompt of the Location Status business process.</td>
<td>System advances to the Status prompt.</td>
</tr>
<tr>
<td></td>
<td>Enter a valid status at the Status prompt.</td>
<td>System returns to the Location prompt.</td>
</tr>
</tbody>
</table>
Admin Lab Exercise 16: Diagnostic Caching

Introduction

This exercise consists of the following activities:

1. Enable Diagnostic Caching

Background Information

In a previous exercise, you learned how to review the contents of the HighJump platform log, and you learned how to elevate the log levels. Diagnostic caching is another method of capturing additional log inventory. Diagnostic caching allows you to temporarily store log 4 and log 5 messages in memory and then write them to the database when certain errors occur in the system.

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the Diagnostic Cache menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Enable Diagnostic Caching

**Background:** Suppose you have a critical, nagging communication-related problem that sporadically appears twice a day. You have reviewed the current logs, and they are not sufficient to identify the root of the problem. You want some elevated logging information to assist with the debugging process. However, because of the negative impact upon performance, you do not want to run the system at log level 5 for an extended period of time. And due to the sporadic nature of the problem, you do not know when to elevate the log level. This presents a dilemma. You want the log level 5 information, but you do not know when to start it.

The concept of diagnostic caching offers a solution to this dilemma. When diagnostic caching is enabled, the system stores a configurable number of severity level 4 and severity level 5 messages in memory. Then, when the system encounters a system log message with a severity level of 1 (critical error), the system automatically flushes these messages from the memory cache and writes them to the log. By using this feature, you get the elevated log information, but only for the time period leading up to the critical error.

While there is some performance degradation when the diagnostic caching feature is enabled, the degradation is significantly less than perpetually running the system in system log level 5 for a single device. Because this feature is heavily dependent upon memory, you should ensure that you have sufficient memory available before you enable the caching.

**Situation:** A couple times a day the Shipping business process encounters critical system error. You want to capture some elevated log information around that event. However, you don't want to run in an elevated log level all day long. You decide to enable diagnostic caching to capture the additional log information.

**Set Up:** In reality, the Shipping business process in the base application works without error. In order to simulate the critical error situation above, you will intentionally introduce a critical failure into the system. This activity uses a package to introduce the failure point. Follow the procedures below to introduce a critical system error into the Shipping business process.

- Open *Advantage Architect*, if necessary.

The next couple steps direct you to import and compile and activate the package in Advantage Architect. However, they do not provide detailed instructions. For additional details on how to import, compile, and activate a package, please see the exercise titled *Exercise 13: Packages*.

- Import the *advantage_architect_critical_failure_release_16.pkg* package in the *Releases\Release 16* … folder on the desktop. (see previous exercises for additional instructions)

- Compile and activate the *WA application* with the *Critical Failure* package. (see previous exercises for additional instructions)

**Current State:** In the steps above, you intentionally introduced a critical failure into the Shipping business process. The steps below demonstrate how the defect manifests itself under the standard logging mechanism.
- Open the **Application Status Console**.
- Open the **System Status Console**.
- Log into the Virtual Terminal as **AMY / AMY / FAAMY** to display the main menu.
- Press the **F8** function key to display the next page of menu options.
- Enter **6** (Load / Ship) at the **OPTION** prompt.
- Enter **2** (Load Shipping) at the **OPTION** prompt.

The system displays the LOAD ID prompt and defaults the value.

![Load Shipping](image1)

- Press the **Enter key** to accept the defaulted **load id**.

The system displays the Carrier Name prompt and defaults the value.

![Load Shipping](image2)

- Press the **Enter key** to accept the defaulted **carrier name**.
The system displays the Pro Number prompt.

Enter ABC at the Pro Number prompt.

The system displays an error message on the prompt screen.

At this point the system has encountered a critical failure. More specifically, it has encountered a system error with a severity level of 1. You introduced this error when you imported, compiled, and activated the package. Consequently, the system writes an entry to the HighJump platform log and displays the log message in the system status console. The status console displays these messages in red because they are severity level 1 messages. Although you cannot see the severity level through the console window, you would be able to see it if you viewed the log messages through Advantage Commander. The graphic below shows the how the log message should appear in the system status console.
Even though it was a critical error, the system only displays a single log messages. If this information is not enough to determine the root cause of the problem you may opt to enable the diagnostic caching feature.

**Procedures:** The procedures below demonstrate how to enable the diagnostic caching feature in order to capture additional log messages that led up to the critical failure.

- Open the **HighJump One Platform UI**, if necessary.
- Click the **Menu | Advantage Commander | Advantage Commander | Platform Configuration | Control Options** menu.

The system displays a list of enterprise-wide control parameters.

<table>
<thead>
<tr>
<th>#</th>
<th>String Key</th>
<th>String Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Adv_url</td>
<td><a href="http://ADVSERVER/Docimg/Adv.png">http://ADVSERVER/Docimg/Adv.png</a></td>
</tr>
<tr>
<td>2</td>
<td>DatabaseBackupFolder</td>
<td>D:\DBBackup\</td>
</tr>
<tr>
<td>3</td>
<td>DialogWordWrapMode</td>
<td>Default</td>
</tr>
</tbody>
</table>

There are several control options that dictate the behavior of the diagnostic cache feature. The chart below gives a brief description of a couple of the settings.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogAppDiagCacheOption</td>
<td>Indicates whether the Diagnostic Cache feature for application messages is enabled or disabled.</td>
</tr>
<tr>
<td>LogAppDiagCacheMaxRecords</td>
<td>The maximum number of application diagnostic log messages to cache for all processes. Valid values are from 1,000 through 100,000.</td>
</tr>
<tr>
<td>LogAppDiagCacheDupDelay</td>
<td>The number of minutes HighJump Advantage Platform waits before saving all cached application messages to the log for duplicate application error messages from the same process.</td>
</tr>
<tr>
<td>LogSysDiagCacheOption</td>
<td>Similar to above. Specific to system log messages.</td>
</tr>
<tr>
<td>LogSysDiagCacheMaxRecords</td>
<td>Similar to above. Specific to system log messages.</td>
</tr>
</tbody>
</table>
Earlier in this manual you learned about two different types of messages that are written to the HighJump log: application messages and system messages. In a similar fashion, there are two sets of options for diagnostic caching. One set of options controls diagnostic caching for the application messages and critical application-type failures. The other set of options controls diagnostic caching for system messages and critical system-type failures. The above table reflects this dual nature of the diagnostic caching feature. The package which you imported and the control options which you will modify relate to the system messages and a system-type critical failure.

- Click the number to the left of the LogSysDiagCacheOption entry.
- Enter 1 in the String Value edit box.
- Click the Update button in the Action Bar.

The system updates the database with the new value, and then returns to the web page which displays a list of all control options.

At this point, you have enabled the diagnostic cache feature. However, the change may not take effect immediately. Because the system does not poll for these control changes every second, you may need to wait up to two minutes before your change takes effect.

- Wait for approximately 2 minutes.

When your changes take effect, the system displays a message in the system status console that indicates the new values of the related parameters.
**Validation:** In the steps above, you manually enabled the diagnostic caching feature. The next several steps demonstrate how the feature works in conjunction with a critical failure.

- Press the **F1** key on the Virtual Terminal until you return to the **USER ID** prompt.

As you navigate backwards through the prompting sequence, the system does not encounter any critical system errors. The system stores the log 4 and log 5 messages in memory. However, it does not write the messages to the log.

- Log into the Virtual Terminal as **AMY / AMY / FAAMY** to display the main menu.
- Press the **F8** function key to display the next page of menu options.
- Enter **6** (Load / Ship) at the **OPTION** prompt.
- Enter **2** (Load Shipping) at the **OPTION** prompt.
- Press the **Enter key** to accept the defaulted **load id**.
- Press the **Enter key** to accept the defaulted **carrier name**.

As you navigate through this prompting sequence, the system does not encounter any critical system errors. The system stores the log 4 and log 5 messages in memory. However, it does not write any messages to the log.

- Enter **ABC** at the **PRO Number** prompt.

After you enter the pro number, the system encounters a critical system error. Because you enabled diagnostic caching on the system side, the system flushes all of the log 4 and log 5 system messages in memory, and writes them to the HighJump platform log table. This is literally several thousand new log messages. The system displays a minimal number of these messages in the system status console. The bulk of the messages can be viewed through Advantage Commander. The system status console should look similar to the following graphic.

![System Status Console](image-url)
Next you will view the flushed messages in Advantage Commander,

☐ In the HighJump One Platform UI, click the Menu | Advantage Commander | Log | Summary Log Report menu.

The system displays the Summary Log Search Criteria page.

☐ Click the Submit button in the Action Bar.

The system displays the Log Message Summary View page.

Also, note the significant number of pages included in the log. The log contains not only the critical error, but also all of the elevated log information in the events leading up to the critical failure.
In Advantage Commander there were two sets of Diagnostic Caching settings: one for the system side and one for the application side. When you enabled the feature, you only enabled it on the system side. As a result, no additional messages were written to the application status console.

By using diagnostic caching, you can review the messages in the moments leading up to the critical error. This will give you a better idea of the events that may have contributed to the critical failure.

**Follow Up:** In the steps above, you enabled diagnostic caching and you witnessed how it impacted the log messages. However, now that you have captured the desired information, there is no need to keep the system in a diagnostic cache mode. Follow the steps below to disable diagnostic caching and to restore the original state of the Shipping process.

- Open the **HighJump One Platform UI**, if necessary.
- In the **HighJump One Platform UI** click the **Menu | Advantage Commander | Advantage Commander | Platform Configuration | Control Options** menu.

The system displays a list of enterprise-wide control parameters.

- Click the number to the left of the **LogSysDiagCacheOption** entry.
- Enter 0 in the **String Value** edit box.
- Click the **Update** button in the Action Bar.
The system updates the database with the new value and then returns to the web page which displays a list of all control options.

<table>
<thead>
<tr>
<th></th>
<th>LogSysDiagCacheDisplay</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>LogSysDiagCacheDupDelay</td>
<td>60</td>
</tr>
<tr>
<td>30</td>
<td>LogSysDiagCacheMaxRecords</td>
<td>10000</td>
</tr>
<tr>
<td>31</td>
<td>LogSysDiagCacheOption</td>
<td>0</td>
</tr>
</tbody>
</table>

At this point, you have disabled the system diagnostic cache feature. However, the change may not take effect immediately. Because the system does not poll for these control changes every second, you may need to wait up to two minutes before your change takes effect.

☐ Wait for approximately 2 minutes.

When your changes take effect, the system displays a message in the system status console that indicates the new values of the related parameters.

The next step directs you to rollback the package that you imported into Advantage Architect. For additional details on how to perform this step, please see the exercise titled Exercise13: Packages.

☐ In Advantage Architect, rollback the Critical Failure package that you previously imported. (see previous exercises for additional instructions)

☐ Compile and Activate the WA application. (see previous exercises for additional instructions)

At this point you have disabled the Diagnostic Caching feature. And you have restored the application to its original state.
Admin Lab Exercise 17:  
Other Administrative Tools

Introduction

This exercise consists of the following activities:

1. View the Log Entries in the Event Viewer
2. Export the Log Entries in the Event Viewer
3. Monitor Real-Time Counters with Performance Monitor
4. Write Performance Monitor Counters to a Log

Background Information

This exercise demonstrates other tools that you have at your disposal for the purpose of administering the HighJump system. The tools in this section reside outside the HighJump suite of products.

Event Viewer is a Windows administrative tool that maintains logs about program, security, and system events the servers. You can use Event Viewer to view and manage the event logs, gather information about hardware and software problems, and monitor Windows security events. The Event Viewer also maintains a log related to the Advantage Platform.

The Performance Monitor is a Windows utility that provides a way for system administrators to collect and monitor a variety of performance-related counters (statistics) on a given server. Sample counters include memory usage, cache usage, network packets per second, and CPU usage.

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the Event Viewer menu and the Performance Monitor menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: View the Log Entries in the Event Viewer

Introduction: The Event Viewer is a Windows administrative tool that maintains logs about program, security, and system events the servers. It also maintains a log related to the Advantage Platform. These logs can provide useful information when troubleshooting HighJump-related problems as well as problems that are outside the HighJump realm. However, it can also be a source of meaningless nonsense. In either case, it is important to know how to access the tool and how to navigate inside it.

Scenario: The warehouse supervisor calls you on the phone. He indicates that the Crystal reports are no longer printing on the printers. You review the HighJump logs and you see that the Event Manager (the HighJump program responsible for printing the reports) shut down unexpectedly. Since the solution is not clear from the HighJump logs, you want to view the logs in the Event Viewer to see if they contain any related information.

Procedures: The procedures below demonstrate how to View the log entries in the Event Viewer. However, they do not reproduce the problem related to the Crystal reports.

☐ Choose the Start | All Programs | Administrative Tools | Event Viewer.

The system opens the Event Viewer tool and presents several menu options in the left panel.

☐ Expand the Applications and Services Logs node.

☐ Click the Advantage Platform Log node.

The system displays a list of all the log entries in the Windows-based Application Platform log.
Use the **vertical scroll bar** to **browse** the log entries in order to get a broad overview of the applications that log events.

Highlight the **first event** in the list.

The system displays the details of the event in the lower portion of the screen.

Click the **down arrow key** several times to view the details of several log entries.
Activity 2: Export the Log Entries in the Event Viewer

Introduction: Even if the log entries in the Event Viewer contain information relevant to the problem at hand, it may require the assistance of another party in order to interpret the data. In these cases, you will need to export the log and send it to the other party.

Situation: This scenario builds upon the one in the previous activity. You are troubleshooting a problem with the Crystal reports. You review the log entries in Event Viewer. You see some entries that may be relevant, but you want to send the logs to the HighJump support team for their assistance.

Procedures: The procedures below demonstrate how to export the log entries out of the Event Viewer. The procedures use the Application log as an example. However, as you work through various production issues, you may want to consider the other logs as well. The procedures do not replicate the problem related to the Crystal reports.

☐ Open the Event Viewer, if necessary.

☐ Expand the Applications and Services Logs node.

☐ Right-click the Advantage Platform Log node.

☐ Choose the Save All Events As menu option.

☐ Click the Desktop menu option in the left panel.

☐ Type advantage_platform_event_log in the File name edit box

☐ Click the Save button.

The system displays a window related to viewing the log on other computers

☐ Click the OK button.

The system creates a file on the desktop. At this point you could zip the file, if necessary, and then send the file to the HighJump support team for further analysis.

☐ Choose the File | Exit menu.
Activity 3: Monitor Real-Time Counters with Performance Monitor

Introduction: Performance Monitor is a Windows tool that allows you to monitor the utilization and operation of a variety of system resources. For example, if performance is slow, you may want to track the CPU usage of the application server. The tool allows you to monitor real time metrics as well as write the metrics to a log file. The metrics are called counters within the tool. Additionally, you can configure Performance Monitor to take action when a specific metric reaches a given threshold.

This activity utilizes a template of predefined metrics provided by the HighJump support team.

Scenario: The warehouse supervisor calls you on the phone. He indicates that there is an overall performance lag across all of the RF readers. You look in the HighJump logs, and you don't see anything indicative of a performance problem. You decide that you want to look at several memory-related metrics on the application server.

Procedures: The procedures below demonstrate how to view several memory-related metrics through the Performance Monitor. However, they do not reproduce the performance problem mentioned in the scenario.

☐ Choose the Start | All Programs | Administrative Tools | Performance Monitor menu.

The system starts the Performance Monitor tool.

☐ Click the Performance Monitor node in the menu panel on the left.

The system immediately begins to track the real-time values for the %Processor Time counter. A counter is a measurement taken on the server. The tool has the ability to track a multitude of counters over a broad range of categories. Some of the categories include: memory, physical disk managements, and the cache.
While the %Processor Time may add some value, you are interested in several of the memory-related metrics. Follow the procedures below to add the memory-related metrics to the graph.

- Click the green cross in the tool bar at the top of the graph. 

The system presents a list of several counter categories.

- Expand the Memory node in the Category list.

The system displays a list of all counters included in the Memory category.

- Click the Memory title in order to highlight all of the memory-related counters.

- Click the Add >> button.
The system adds the Memory counter to the right side of the window.

Click the OK button.

The system begins to monitor and chart the new counters.

If you are familiar with the meaning of any of these metrics, then the graph will provide valuable information for troubleshooting the problem. If you are not familiar with the meaning of the metrics, then you may want to write this information to a log and pass it to another party. You will learn how to write the graph information to a log in a later activity.
Activity 4: Write Performance Monitor Counters to a Log

Introduction: The previous activity demonstrated how to monitor various real-time counters. However, the system did not store the data on the hard-drive. Rather, it maintained about 90 seconds of data in the graph before it was overwritten. There may be times when you want Performance Monitor to capture several hours of counter data. In that case, you need to write the counter values out to a log. This collection of data that is written to a log is called a Data Collector Set.

Scenario: This scenario builds upon the one from the previous activity. You review the real-time memory counters in Performance Monitor. You don’t see anything that would contribute to the overall performance problems of the RF readers. You call the HighJump support team. They recommend that you use Performance Monitor to capture 2 hours of basic metrics, and then send the log to them.

Procedures: The procedures below demonstrate how to write the counter data, and several other items to a log. However, they do not reproduce the performance problem mentioned in the scenario.

☐ Create a new folder on the desktop called **Performance Monitor Output**

☐ Open the **Performance Monitor**, if necessary.

☐ Expand the **Data Collector Sets** node in the panel on the left.

☐ Right-click the **User Defined** node.

☐ Choose the **New | Data Collector Set** menu option.

The system opens a wizard which allows you to define several properties of the information which will be written to the log.

![Create new Data Collector Set](image)

The Name property is an identifier which relates to the entire set of information (the Data Collector Set) which the system will monitor and store in a log.

☐ Enter **HighJump** in the **Name** edit box.
The system gives you the ability to configure which counters to include in the log as well as several other attributes. However, in general, building a Data Collector Set from a predefined template is an easier way to start. The Performance Monitor tool includes a basic template which can be used. The procedures below demonstrate how to build a Data Collector Set from the system template.

- Select the **Create from a template** radio button.
- Click the **Next** button.
- Select the **Basic** option in the **Template Data Collector Set** box.
- Click the **Next** button.

The system displays a window which allows you to configure the folder which will hold the logging information. You can place the folder anywhere on the drive. For simplicity sake, you will place the logging information in a new folder on the desktop.

- Click the **Browse** button.
- Expand the **Desktop** node.
- Select the **Performance Monitor Output** folder.
☐ Click the **OK** button.

The system displays the modified root directory in the edit box.

☐ Click the **Next** button.

☐ Click the **Finish** button.

The system displays the new HighJump Data Collector set in the tree structure on the left and in the panel on the right.

You have created the basics of the data collector set. However, there are several additional properties you should set prior to capturing the data. In addition to the basics, you must also specify which counters you want the system to track.

☐ Expand the **User Defined** node in the left panel.

☐ Click the **HighJump** node.

☐ Right-click the **Performance Counter** option in the right panel.

☐ Choose the **Properties** menu option.

☐ Click the **Add** button.

☐ Click the **Process** entry in the left panel.
Even though there is no change to the interface in top portion of the window. The system shows a list of all Process-related counters in the bottom portion of the window.

![Available counters](image1)

- Click the `<All Instances>` entry in the bottom portion of the window.
- Click the Add >> button.

The system adds all of the Process-related counters (noted by the *) to the Added Counters panel on the right side of the window.

![Added counters](image2)

- Click the OK button.

The system, once again, shows that all Process-related counters will be tracked and added to the log.
In addition to configuring the counters that you want to capture, the Performance Monitor also gives you the ability to specify the format of the log and how often the system should take a reading of each of the counters. In this example, you will specify a comma-delimited format and an interval of 10 seconds between each sample reading.

- Choose Comma Separated in the Log Format drop-down list.
- Choose 10 in the Sample Interval selector box.
- Click the Apply button.

The settings should look like the following graphic.

In addition to configuring the counters, the format, and the interval, Performance Monitor also allows you the ability to specify the filename of the log. This can be useful, if you intend to perform multiple captures over the span of a couple days. In this example you will specific a filename that includes the computer name and a date stamp embedded in the name.

- Click the File tab along the top of the window.
- Enable the Prefix file with computer name checkbox.
- Click the > button to the right of the File name format edit box.
- Choose the yyyy menu option.

The system adds yyyy to the File name format edit box.
Admin Lab Exercise 17:
Other Administrative Tools

Click the > button to the right of the File name format edit box.

Choose the MM menu option.

The system adds MM to the File name format edit box.

Click the > button to the right of the File name format edit box.

Choose the dd menu option.

The system adds dd to the File name format edit box.

Click the Apply button.

Click the OK button.

Performance Monitor also gives you the ability to specify a schedule and stop conditions which control when the system will capture the data. In this example, you will omit the schedule, which will force a manual start. And you will instruct the system to continue gathering the data indefinitely. This will force a manual stop.

Right-click the HighJump node in the left panel.
Choose the **Properties** menu option.

Click the **Stop Condition** tab.

Disable the **Overall Duration** checkbox.

The settings should look like the following graphic.

![HighJump Properties](image)

Click the **Apply** button.

Click the **OK** button.

At this point, you have configured all of the essential options. The next step is to run the Data Collector Set, which will start collecting the data.

Right-click the **HighJump** node in the left panel.

Choose the **Start** menu option.

Wait **1 or 2 minutes** while the system collects some data.

Right-click the **HighJump** node in the left panel.

Choose the **Stop** menu option.

The system captured the log information and placed it into the folder on the desktop.

Open the **Performance Monitor Output** folder on the desktop.

The system has created a subfolder. The name of the subfolder is the machine name followed by a date stamp in the format of YYYYMMDD followed by a sequential number.
At this point you could zip the contents of the folder and send it to the HighJump Worldwide Support Team for their analysis.
Admin Lab Exercise 18: Schedule a Process Object

Introduction

This exercise consists of the following activities:

1. Technical Overview
2. Schedule an Inventory Snapshot

Background Information

Often times, the HighJump system needs to run a business process at a specific time of the day. Often times, these business processes have no user interface. For example, you may need to generate Cycle Count tasks every day at 0300 hours or you may need to print an inventory report based upon the HighJump tables every day at 0700 hours.

The HighJump system does not contain any inherent scheduling mechanism to launch a business process at a specific time. Rather, you must use the tools in the operating system to schedule a business process.

You have already run several business processes in this class. You have initiated most of the business processes through the Virtual Terminal or the Web Terminal. You have also initiated a couple through pages in the HighJump One Platform UI. The HighJump system includes a third method for launching business processes. The third method is through a stand-alone executable that can be called from a command line. The stand-alone executable is specifically geared towards the scheduled, unattended (no user interface) business processes. The activities in this exercise utilize the stand-alone executable.

Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Schedule Process Objects menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Technical Overview

Since the HighJump system does not contain any scheduling mechanism, you must leverage the tools in the operating system. When you define a scheduled task in the operating system, you provide two basic types of information: the schedule (time and frequency) and the executable name with its parameters. The time and the frequency are self-evident. The stand-alone executable requires some further explanation.

The HighJump platform includes a stand-alone executable called RUNPROCESSOBJECT.EXE. The purpose of this executable is to run a process object from a command line. A process object, also called a workflow or a business process, is basically a procedure developed in an Advantage Architect application. The executable allows you to initialize fields (variables) to specific values and pass them into the process object. You will point to this executable when you define the scheduled task in the operating system.

The syntax and an example are below.

RUNPROCESSOBJECT.EXE [AppName].[Process Object Name] [Field 1 Name] [Field 1 Value] [Field 2 Name] [Field 2 Value]…

D:\HJS\Adv\Bin\RunProcessObject.exe "WA_Export Receipts" "SYS_LOCALE" "101"

RunProcessObject.exe "application name.process object name" "first field name" "first field value"

The application name and the process object name are separated by a period. Each name or value is separated by a space character. As with most command line driven programs, parameters that include spaces must be surrounded by quotes.
Activity 2: Schedule a Process Object

Background: The WA application in Advantage Architect contains a process object that takes a snapshot of the WA inventory table and sends it to the host. The WA inventory is stored in a table called t_stored_item. The snapshot of the inventory is temporarily stored in a table called t_al_host_inventory_snapshot. The name of the process object that performs the snapshot is “App>Msg Inventory Snapshot”

Scenario: You want the HighJump system to take a snapshot of its inventory in warehouse 01 and send it to the host system every day at 0300 hours. After the host system accepts the inventory snapshot from WA, it compares the inventory levels between its record and the data sent from WA. Then it generates a report on the discrepancies.

Procedures: The procedures below demonstrate how to schedule the snapshot process object to run at 0300 hours. The host component is not configured in the training environments. The process object will take the snapshot, but it will not be able to send the data to the host.

☐ Open the Microsoft SQL Server Management Studio program and log into it.

☐ Choose the File | New | Query with Current Connection menu.

☐ Choose AAD in the drop-down list near the upper left corner.

☐ Type the following text into the query window

```
select * from t_al_host_inventory_snapshot
```

☐ Choose the Query | Execute menu.
The system shows the current contents of the snapshot table. In this case, the snapshot table is empty. After you schedule and run the process object, you will run this query a second time, and the table will contain several rows of data.

<table>
<thead>
<tr>
<th>inventory_snapshot_id</th>
<th>host_group_id</th>
<th>transaction_code</th>
<th>item_number</th>
<th>lot_number</th>
<th>quantity</th>
</tr>
</thead>
</table>

☐ Choose the Start | All Programs | Administrative Tools | Task Scheduler menu.

The system opens the main window of the Task Scheduler tool.

☐ Choose the Action | Create Basic Task menu option.

This system opens a wizard which will allow you to create and schedule a task.

The first item you will provide is a name for the task. The Name will serve as an identifier for the task. In this example, Inventory Snapshot is a good identifier for the task.
☐ Enter **Inventory Snapshot** in the **Name** edit box.

☐ Click the **Next** button.

The next settings you must configure relate to how often you want the task to run. In this example you will schedule the task to run every day at 03:00. Then for testing purposes you will initiate it manually.

☐ Select the **Daily** radio button.

☐ Click the **Next** button.

☐ Set the **Calendar** drop-down to **tomorrow’s date**.

☐ Set the **Time** control to **03:00:00 AM**.

The settings should look similar to the following graphic.

![Schedule settings](image)

☐ Click the **Next** button.

The next settings you must configure relate to which program you want to run. In this example, you want to run a command-line program called RunProcessObject.exe. RunProcessObject.exe is a program written by HighJump Software which allows you to run a process object from a source outside of the HighJump system.

☐ Select the **Start a Program** radio button.

☐ Click the **Next** button.

☐ Click the **Browse** button.

☐ Click the **Applications (D:)** drive.

☐ Navigate to the **Program Files | HighJump Software | Adv | Bin** folder.

☐ Click the **RunProcessObject** option.

☐ Click the **Open** button.
The setting should look like the following graphic.

Additionally, you need to specify which process object you want to run and how to initialize the fields (variables). In this example, the name of the process object is App>Msg Inventory Snapshot which resides inside the WA Processors application. And you need to initialize the Warehouse ID field to a value of 01.

☐ Enter the following string in the Add Arguments (optional) edit box. (You must include the double quotes.)

"WA Processors.App>Msg Inventory Snapshot" "Warehouse ID" "01"

The setting should look like the following graphic.

☐ Click the Next button.

The system then summarizes the settings.
Click the Finish button.

At this point, you have configured all of the settings for the task. Now it is time to test. Since you configured the task to run every day at 0300 hours, you could wait until tomorrow morning to see if the task ran successfully. Or you can run the task manually. In this example, you will initiate the task manually.

Click the Task Scheduler Library entry in the panel on the left.

The system shows an entry for the task you just created along with several other pre-existing tasks.

Right-click the Inventory Snapshot entry.

Choose the Run option.

The system shows a black popup window briefly, and then the window disappears. This is the only visual indication that the task ran correctly. However, if you refresh the screen, you will see that the system updates the Last Run Time and the Last Run Result attributes of the task in the main screen.
☐ Click the **Refresh** entry in the panel on the right.

☐ Use the **horizontal scroll bar** to view the columns on the right.

The system indicates that the task successfully completed.

![Table showing successful task results](image)

The Task Scheduler successfully ran the task. But the real test is whether or not the system populated the database table with the data that needs to go to the host. In this example, you will run the same script that you ran at the top of this activity.

☐ In **Microsoft SQL Server Management Profile**, choose the **Query | Execute** menu.

The database table which was empty at the top of this activity, now contains the data that needs to be sent to the host. The process object associated with the task ran successfully and it populated the table.

![Database table showing populated data](image)
Admin Lab Exercise 19: SQL Server Profiler

Introduction

This exercise consists of the following activities:

1. Import an SQL Profiler Template
2. Run an SQL Profiler Trace
3. Modify an SQL Profiler Trace
4. Export an SQL Profiler Trace

Background Information

SQL Profiler is a Microsoft tool that allows you to monitor an instance of a Microsoft SQL Server Database engine. Within the tool you can create a trace which captures database events based upon configurable parameters. This tool is commonly used to identify long-running database queries. (Oracle has a similar profiling component; however it is not covered in this exercise.)

There is some overhead consumed by the SQL Profiler. Depending on the scope of the trace, you may experience a bit of a performance hit on the readers. While this tool can be helpful in identifying long-running database queries and other database-related problems, you should turn it off after you have captured the desired information.

Before performing this exercise, we recommend you use the Troubleshooting Advantage System Pre-Work link and review the related video demonstrations under the SQL Profiler menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Import an SQL Profiler Template

**Background:** When you run a trace in SQL Profiler, you must first define the properties of the trace. (You will learn how to do this in a later activity.) Once you have defined the properties of the trace you can save the definition as a template. An SQL Profiler template is a pre-configured set of trace properties. This template can then be used as a basis for any future traces. Generally speaking, it is easier to run a trace from a pre-defined template than it is to create a trace from scratch.

The sample template used in this activity is specific to the training exercise. It will not work very well in a production environment. If you want to use a template as a basis for your trace, then you should either create your own template or request one from the HighJump support team.

**Scenario:** Shortly after the original go-live event, the shipping business process begins to show some significant performance problems on the terminals. The shippers are sometimes waiting up to 15 seconds between screens. Your normal course of action is to contact the database administrator at your company. However, he is on vacation. You opt to contact the HighJump support team. After a couple minutes of initial diagnostics, the HighJump support representative sends you an SQL Profiler template via e-mail. She asks you to run an SQL Profiler trace based upon the template she provided.

**Procedures:** The procedures below demonstrate how to import an SQL Profiler template with a template. However, they do not reproduce the performance problem in the shipping process.

☐ Choose the **Start | All Programs | Microsoft SQL Server 2014 | SQL Server 2014 Profiler** menu.

The system opens the SQL Server Profiler tool. Initially, it displays a grey window with menu options along the top.

Before you can use a template, you must first import the template into the database.

☐ Choose the **File | Templates | Import Template** menu.

The system opens the standard dialog for selecting a file.
☐ Click the Desktop menu option on the left panel.
☐ Navigate to the Training \ Admin \ Profiler folder.
☐ Click the hjs_go_live_profiler_sql_2014 file.
☐ Click the Open button.

The system displays a dialog indicating that the import was successful.

☐ Click the OK button.

Now that the template has been imported into the database you can create a trace which uses the template as a basis. This is the focus of the next activity.
Activity 2: Run an SQL Profiler Trace

Background: In the previous activity you imported a template into the database. In this activity you will use that template as a basis for a new trace.

Scenario: This scenario is identical to the one from the previous activity.

Procedures: The procedures below demonstrate how to start an SQL Profiler trace. However, they do not reproduce the performance problem in the shipping process.

The HighJump engine does not need to be running in order to use SQL Server Profiler. However, if you want to capture data related to the HighJump engine, then you must start the engine.

☐ Start the HighJump engine, if necessary.

☐ In SQL Server Profiler, choose the File | New Trace menu.

The system opens the connection window that you have seen in conjunction with the Microsoft SQL Server Management Studio tool.

☐ Enter the machine name of the database server in the Server Name edit box.

☐ Use the SQL Server section of the login.txt file to log into the Profiler tool.

The system opens a window which allows you to define the properties of the trace.
Enter **Shipping Performance** in the **Trace Name** edit box.

Choose **hjs_go_live_profiler_sql_2014 (user)** in the **Use the Template** drop-down list.

This template is the one that you imported in the previous activity. If you didn’t want to use a template, you would not change the value in the drop-down list.

The properties of the trace have already been defined because the trace has been built upon a template. At this point, there is no need to modify the trace definition.

Click the **Run** button.

The system immediately begins to display events in the window. Even though there are no users on the system, there is a significant amount of activity in the database. This activity results from several processes that run in the background.
After you have captured the desired information, you will need to stop the trace. You will learn how to store this information to a file during a later activity.

- Choose the File | Stop Trace menu in the SQL Profiler.
- Select any of rows in the main screen in order to highlight the row.

The system displays the SQL statement in the bottom panel.

- Use the up and down arrows to view the contents of several statements.
Activity 3: Modify an SQL Profiler Trace

Scenario: This scenario builds upon the one from the previous activity. You take a couple screen shots of the previous trace and send them to HighJump. After she reviews the information, the HighJump support member has some suspicions about the root cause of the problems in the ship process. However, she would like to capture some additional information in order to confirm her suspicions.

She asks you to run a similar trace, but to make the following additions:
1. Capture the Error Number and the Host Name attributes for each event
2. Capture the SQL Transaction events
3. Capture only the events from the AAD database

Procedures: The procedures below demonstrate how to modify an SQL Profiler Trace. However, they do not reproduce the performance problem in the shipping process.

☐ Choose the File | Properties menu.

The system displays the same window in which you entered the trace name and the template name.

The next several steps demonstrate how to capture the Application Name and the Host Name attributes for each event.

☐ Click the Events Selection tab.

The system opens a window that allows you to configure which events and columns are used in the trace.
Fill in the blanks with the correct information:

Disable the **Show all Events** checkbox in the lower right corner of the window.

Enable the **Show all Columns** checkbox in the lower right corner of the window.

Use the **horizontal scroll bar** to scroll to the right.

The columns with a white background indicate that at least one event displays the given data element. A marked checkbox indicates that the given event displays the given data element in the trace. The columns with the gray background indicate that none of the events display the given data element.

Right-click the header of the **Error column**.

Choose the **Select Column** menu option.

The system enables all of the checkboxes in the Error column.
Right-click the header of the HostName column.

Choose the Select Column menu option.

The system enables all of the checkboxes in the Host Name column.

Disable the Show all Columns checkbox in the lower right corner.

The system hides all of the unused columns, and it only displays the ones that are included in the trace. It appends the Error and the HostName columns to the end of the existing list.

At this point you could continue with the remainder of the changes. However, for the purpose of demonstration, this activity immediately tests the changes.
☐ Click the **Run** button.

Like before, the system immediately begins to display the events.

<table>
<thead>
<tr>
<th>SPID</th>
<th>TextData</th>
<th>StartTime</th>
<th>EndTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>SELECT CASE WHEN CONVERT ( FLOAT, ...</td>
<td>2011-03-17 11:26:28...</td>
<td>2011-03-17</td>
</tr>
<tr>
<td>51</td>
<td>SET ROWCOUNT 0</td>
<td>2011-03-17 11:26:28...</td>
<td>2011-03-17</td>
</tr>
<tr>
<td>51</td>
<td>SELECT isnull( next_value , 0 )</td>
<td>2011-03-17 11:26:28...</td>
<td>2011-03-17</td>
</tr>
</tbody>
</table>

☐ Choose the **File | Pause Trace** menu in the SQL Profiler.

☐ Use the **horizontal scroll bar** to scroll all the way to the right.

The system now displays the Error number and the Host Name for each event in the far right columns.

The next several steps demonstrate how to capture the SQL Transaction events.

☐ Choose the **File | Properties** menu.

☐ Click the **Events Selection** tab.

☐ Enable the **Show All Events** checkbox in the lower right corner.

☐ Disable the **Show all Columns** checkbox in the lower right corner of the window.
The system displays a list of categories from which you can select the desired events. The categories in bold text (like Stored Procedures and TSQL) indicate that at least one of the events in the category is selected for monitoring.

Expand the **Transactions** category

Enable the **SQL Transaction** checkbox

The system automatically enables the entire set of attributes associated with the event. Whenever the system detects an SQL Transaction event, it will log the event to the trace and displays the associated SPID, Text Data, Start Time, and the other selected attributes.

Disable the **Show All Events** checkbox.

The system only displays the events it will capture. This list now includes SQL Transactions in the Transactions category.
At this point you could continue with the remainder of the changes. However, for the purpose of demonstration, this exercise tests the work you have performed so far.

- Click the Run button.

Like before, the system immediately begins to display the events.

<table>
<thead>
<tr>
<th>SPID</th>
<th>TextData</th>
<th>StartTime</th>
<th>EndTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>SELECT CASE WHEN CONVERT ( FLOAT,...</td>
<td>2011-03-17 11:26:28...</td>
<td>2011-03-17</td>
</tr>
<tr>
<td>51</td>
<td>SET ROWCOUNT 0</td>
<td>2011-03-17 11:26:28...</td>
<td>2011-03-17</td>
</tr>
<tr>
<td>51</td>
<td>SET ROWCOUNT 0</td>
<td>2011-03-17 11:26:28...</td>
<td>2011-03-17</td>
</tr>
<tr>
<td>51</td>
<td>SELECT isnull ( next_value , 0 ) F...</td>
<td>2011-03-17 11:26:28...</td>
<td>2011-03-17</td>
</tr>
</tbody>
</table>

- Choose the File | Pause Trace menu in the SQL Profiler.
- Browse through the result set.

As you browse through the data, you will now see some trace data generated for the SQL Transaction events. And for each SQL Transaction event it lists several different attributes.

The next several steps demonstrate how to apply a filter and only capture those events that take place in the AAD database.

- Choose the File | Properties menu.
- Click the Events Selection tab.
- Click the Column Filters button in the lower right corner.

The system displays the Edit Filter window.
Click the **DatabaseName** entry in the left panel.

Expand the **Like** node in the right panel.

Enter **AAD** in the **Like** edit box.

Press the **Tab** key.

The system places a funnel icon to the left of the DatabaseName entry. From this point forward the trace will only capture events in the AAD database.

Click the **OK** button.

Click the **Run** button.

Choose the **Edit | Clear Trace Window** menu.

The system removes all of the entries from the current trace window and briefly displays an empty window.
After a second or two, the system begins to display the events.

Choose the **File | Pause Trace** menu in the SQL Profiler.

Use the **horizontal scroll bar** to scroll to the right until you see the **DatabaseName** column.

Now the system only captures those events in the AAD database.
Activity 4: Export an SQL Profiler Trace

Scenario: This scenario builds upon the one from the previous activity. You have configured the trace to meet the requirements specified by the HighJump support team. And you have captured a significant amount of information. Now you want to send this information to the HighJump team so they can analyze the data.

Procedures: The procedures below demonstrate how to export the data from an SQL Profiler trace. However, they do not reproduce the performance problem in the shipping process.

☐ Choose the File | Save As | Trace File menu.

The system opens the standard dialog used to save a file on a drive.

☐ Click the Desktop link in the left panel.

☐ Enter shipping_performance_trace in the File Name edit box.

☐ Click the Save button.

The system saves the file to the desktop. At this point, you could zip the file if necessary, and then send it to the HighJump support team for further analysis.

☐ Choose the File | Exit menu.

☐ Click the Yes button.
Admin Lab Exercise 20: Employee Messaging

Introduction

This exercise consists of the following activities:

1. Send a message to an employee.
2. View a response to an employee message.

Background Information

Sometimes an administrator needs to contact an individual employee who is working in a remote corner of the warehouse. Also, sometimes an administrator has a message to communicate to an entire department. Obviously, the phone and the loudspeaker system are options for sending these messages to the desired recipients. However, the HighJump system also contains a method for transmitting messages to employees on RF terminals, Web Terminals or Virtual Terminals. This method is called Employee Messaging.

Employee Messaging is a web-based tool for exchanging messages with an individual on a terminal. It is important to know that the messages sent from the server to the terminals do not automatically appear on the terminal. The logic running on the terminal only checks for messages at certain places during the business processes. It checks for messages when the user exits a business process and returns to the menu. It also checks for messages at some strategic places within the actual business process. This reason alone may be enough to sway an administrator towards the phone or the loudspeaker option.

Before performing this exercise, we recommend you use the Advantage System Administration Pre-Work link and review the related video demonstrations under the Messaging menu. (The link is located on the right side panel of the VLab interface.)
Activity 1: Send a Message to an Employee

Scenario: You have some important information that you want to convey to Amy in person. However, she is working on an RF terminal in a remote corner of the warehouse where there are very few phones. You decide to send a HighJump Employee Message to Amy instructing her to come to your office.

Procedures: The procedures below demonstrate how to send an Employee Message to an individual on a terminal.

☐ Start the **engine**, if necessary.

☐ Start the **Virtual Terminal**, if necessary.

☐ Log into the Virtual Terminal as **AMY**, if necessary.

☐ Navigate to the **Location prompt** of the **Location Status** business process.

☐ Open the **HighJump One Platform UI**, if necessary.

☐ Click the **Menu | Supply Chain Advantage | Advantage Dashboard | Employee Messaging | Send Message to Employee** menu.

The system opens the **Send Message to an Employee** page.

- Choose **01-Warehouse 01** in the **Warehouse ID** drop-down list.

- Choose **AMY ADMINISTRATOR – AMY** in the **Employee** drop-down list.

- Type **come to office, pls** in the **Message** multi-line edit box.

- Choose **High** in the **Priority** drop-down list.
Depending on the type of message, you may want Amy to enter a response instead of simply acknowledging the message with the enter key. The system supports either style. This option is dictated by the Response Required parameter.

- Choose Yes in the Response Required drop-down list.

The window should look like the following screenshot.

![Send Message to an Employee](image)

- Click the Send button in the Action Bar.

The system sends the message to Amy, and it displays an acknowledgement screen.

![Sent Employee Message](image)

**Test Cases:** The final step in this process is to validate that the changes work as expected. In previous activities, the test cases have been outlined in a chart format. However, this activity demands some
additional explanations. As a result, the test cases are outlined in a procedure format instead of a chart. If a given procedure does not yield the expected outcome, then review the steps from the previous activity. It is likely that you either missed a step or that you performed a step incorrectly.

☐ Click on the Virtual Terminal to give it focus.

No messages appear on the Virtual Terminal. This is the expected behavior. The Location Status business process does not check for messages here.

☐ Enter M102 at the Location prompt.

The system advances to the Location Status prompt. However, it does not display any messages. The Location Status business process does not check for messages when the user enters a location.
Box: Enter I at the **Location Status** prompt.

The system changes the status of the location, and then it checks for employee messages. In this case it finds one, so it displays the text and provides an opportunity for Amy to respond to the message.

```
Location Status

come to office, pls

RESPONSE:
-  
```

Box: Enter **OK** at the **Response** prompt.

The system stores Amy's response for later review, and then the Virtual Terminal continues with the location prompt screen.

```
Location Status

Scan the location.

LOCATION
-  
```

As you saw in this example, the system does not display the employee messages to the user in real time. Nor does it check for employee messages after every prompt. It only checks for employee messages at specific places in the business process, and that place may differ between the business processes. If this limitation is acceptable for the type of message you are sending, then Employee Messaging may be a good option for delivering the message to an employee.
Activity 2: View a Response to an Employee Message

Scenario: This scenario builds upon the scenario from the previous activity. It has been several hours since you initially sent the message to Amy’s terminal. However, Amy has not stopped by your office. You want to determine if Amy has seen and acknowledged the message.

Procedures: The procedures below demonstrate how to view a response to an employee message.

1. Open the HighJump One Platform UI, if necessary.
2. Click the Menu | Supply Chain Advantage | Advantage Dashboard | Employee Messaging | View Employee Messages menu.

The system displays the Search for Employee Messages page.

Search for Employee Messages

Warehouse ID
<ANY>
Department
<ANY>

Click the Query button in the Action Bar.

The system displays a list of all employee messages. This page includes the message itself, and as you scroll to the right, you can also see Amy’s response to the message. If Amy had not seen the message on her terminal, then the Employee Response column would be empty.
<table>
<thead>
<tr>
<th>Response Date</th>
<th>Response Time</th>
<th>Employee Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/7/2016</td>
<td>10:36:21</td>
<td>OK</td>
</tr>
</tbody>
</table>