

HP Asset Manager

Using IP Addresses to Set Asset Location in Asset Manager®

Automating Location Assignment



Legal Notices	2
Introduction	3
Prerequisites	3
How to use the solution	3
Assign IP ranges to locations	3
Use a wizard to update the asset's location	4
Use a workflow to update the asset's location	4
How to implement the solution	4
Create the IPRange table	5
Add fields to amIPRange	5
Add links to amIPRange	6
Add an index to amIPRange	7
Add a validity script to amIPRange	8
Add a page to amIPRange	9
Create a screen for amIPRange	9
Add a page to amLocation	10
Modify a screen for amLocation	10
Create a wizard	11
Create the script for the workflow	14
Create the workflow	16
Limitations of the solution	17
Ways to extend this solution	17
Summary	18
For more information	19

Legal Notices

© Copyright 1994-2008 Hewlett-Packard Development Company, L.P.

Confidential computer software.

Valid license from HP required for possession, use or copying.

Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services.

Nothing herein should be construed as constituting an additional warranty.

HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Adobe®, Adobe logo®, Acrobat® and Acrobat Logo® are trademarks of Adobe Systems Incorporated.

Corel® and Corel logo® are trademarks or registered trademarks of Corel Corporation or Corel Corporation Limited.

Java™ is a US trademark of Sun Microsystems, Inc.

Microsoft®, Windows®, Windows NT®, Windows® XP, Windows Mobile® and Windows Vista® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Introduction

Maintaining accurate asset location data can be one of the most challenging tasks in an asset management process. The implementer of asset management technology has the responsibility for employing automation whenever possible to help the organization maintain the most accurate data possible. Many companies use discovery technology to assist in this effort. While discovery tools do a good job of telling you what the computer is and what software it has on it, they generally do a poor job telling you where the computer is in the physical world. This lack of an automated way of collecting critical location data poses a big problem for asset management practitioners.

Knowing an asset's location is critical when managing moves, leases, contracts, and fixed asset accounting; and in Sarbanes-Oxley compliance. To compensate for the lack of tool support for maintaining this data element, many asset management teams roll out manual processes that depend on the proper training and discipline of IT and facilities staff.

Fortunately, by using a combination of discovery technology and Asset Manager®, there is a way to close the automation gap when it comes to maintaining location data. This document describes a way to use the IP information collected by the organization's discovery tool in combination with Asset Manager location data to specify where an asset is physically located.

Prerequisites

To use this solution effectively, the organization must have Asset Manager (4.x or higher) and a discovery tool like Enterprise Discovery. The discovery tool must be able to collect the IP address of computers in the environment. Using a tool such as Connect-It, that IP address information should be imported into Asset Manager (amComputer).

In addition to the IP address of the computer, Asset Manager will need to have a list of the physical locations where the computer may be located. This data is stored in amLocation. Once this foundation of data is established, IP address ranges should be associated with each location.

How to use the solution

Once you have the discovery tool feeding IP address information into Asset Manager and you have physical locations defined, there are at least two ways to correlate the computer's virtual location (its IP address) with a physical location. The first way is to use a wizard to help a technician narrow down the possible physical locations and then assign a location to the asset. The second way is to use a periodically triggered workflow to have Asset Manager assign locations to assets that have an IP address but no location assigned to them. The following procedures will walk through each scenario.

Assign IP ranges to locations

After this solution is implemented, someone will need to assign IP ranges to locations.

1. Navigate to the Locations screen.
2. Click the **IP Range** tab, add a record and enter an IP range start address and an IP range end address. The start and end addresses may be the same. You may associate multiple IP ranges with a single location.

Note: Each IP address is stored as a number to enhance the performance of the solution. Enter range values without decimal points, and with leading zeroes added. For example, for the range 128.1.1.256, enter 128001001256. The resulting value will be formatted to 128,001,001,256.

Use a wizard to update the asset's location

After this solution is implemented and IP ranges are assigned to locations, a user would do the following:

1. Navigate to the Computers screen.
2. Create a filter where Portfolio.Location.IsLocald = 0 and TcpIpAddress <> "".
3. Right-click a computer from the filtered list and click **Actions** -> **Choose location using IP address**. This launches the wizard.
4. Asset Manager lists possible location matches. Select the best match and click **Finish**. Asset Manager updates the portfolio item related to the computer with the location selected.

Use a workflow to update the asset's location

You may prefer to automate this process even more. After this solution is implemented and IP ranges are assigned to locations, you can schedule the workflow included in this solution to evaluate each asset with an IP address but no location, and assign the first matching location. For information about how to schedule a periodic workflow, refer to the Asset Manager product documentation.

How to implement the solution

This solution consists of three main parts:

- Create the IPRange table
- Create a wizard
- Create the workflow

Creating the IPRange table is mandatory. Implementing both the wizard and the workflow provides the most flexibility, but depending on the organization's approach, you may want to implement only one of them.

For organizations that want more control over the location that is assigned, HP Software suggests that you use the wizard. For organizations that want the maximum coverage in the least amount of time, HP Software suggests that you use the workflow. The workflow is particularly effective during the initial population of Asset Manager, when the organization may not have much accurate location data from legacy systems but may have accurate information from discovery tools.

The following sections describe each of the tasks necessary to implement the solution.

Create the IPRange table

The screenshot shows the configuration for the **amIPRange** table and the **dtLastModif** field.

Table Configuration (amIPRange):

- SQL Name: amIPRange
- Label: IP ranges
- Description: IP ranges
- System object: No
- String: {IPRangeld}
- Validity: {Script}
- Relevance: Yes

SQL name	Label	Description	System object	Keep history	Read only	Mandatory
dtLastModif	Modified on	Date of modification	Yes	No	No	No
EndAddress	End address	IP range end address	No	No	No	No
IDHCPServerId	ID of link DHCP server	ID of link DHCP server	Yes	No	No	No
IPRangeld	IPRangeld	IPRangeld	Yes	No	No	No
ILocationId	ID of link Location	ID of link Location	Yes	No	No	No
StartAddress	Start address	IP range start address	No	No	No	No
Status	Status	Status	No	No	No	No

Field Configuration (dtLastModif):

- SQL Name: dtLastModif
- Label: Modified on
- Description: Date of modification
- System object: Yes
- Domain: [Empty]
- Screen set: [Empty]
- Type: Date and time
- Size: 8
- User type: Default
- Format: [Empty]
- Upgrade: [Empty]

1. In the Asset Manager Application Designer, click the **Database** menu, and click **Add a table**.
2. Enter the table information and click **Create**:
 - SQL Name: **IPRange**
 - Description: **IP ranges**
 - Primary key: **IIPRangeld**

Add fields to amIPRange

The screenshot shows the configuration for the **amIPRange** table and the **dtLastModif** field, identical to the previous image.

Table Configuration (amIPRange):

- SQL Name: amIPRange
- Label: IP ranges
- Description: IP ranges
- System object: No
- String: {IPRangeld}
- Validity: {Script}
- Relevance: Yes

SQL name	Label	Description	System object	Keep history	Read only	Mandatory
dtLastModif	Modified on	Date of modification	Yes	No	No	No
EndAddress	End address	IP range end address	No	No	No	No
IDHCPServerId	ID of link DHCP server	ID of link DHCP server	Yes	No	No	No
IPRangeld	IPRangeld	IPRangeld	Yes	No	No	No
ILocationId	ID of link Location	ID of link Location	Yes	No	No	No
StartAddress	Start address	IP range start address	No	No	No	No
Status	Status	Status	No	No	No	No

Field Configuration (dtLastModif):

- SQL Name: dtLastModif
- Label: Modified on
- Description: Date of modification
- System object: Yes
- Domain: [Empty]
- Screen set: [Empty]
- Type: Date and time
- Size: 8
- User type: Default
- Format: [Empty]
- Upgrade: [Empty]

1. In the Asset Manager Application Designer, navigate to the IPRange table.

2. Create the field StartAddress:
 - SQL Name: **StartAddress**
 - Label: **Start address**
 - Description: **IP range start address**
 - Type: **Double-precision number**
3. Create the field EndAddress:
 - SQL Name: **EndAddress**
 - Label: **End address**
 - Description: **IP range end address**
 - Type: **Double-precision number**
4. Create the field Status:
 - SQL Name: **Status**
 - Label: **Status**
 - Description: **Status**
 - Type: **Custom itemized list**

Add links to amIPRange

The screenshot shows the Asset Manager Application Designer interface. The top section displays the configuration for the **amIPRange** table:

- SQL Name: amIPRange
- Label: IP ranges
- Description: IP ranges
- Hierarchy: Can contain features
- System object: No
- String: [IPRangeId]
- Validity: (Script)
- Relevance: Yes

Below this is a table listing related objects:

SQL name	Label	Description	System object	Keep history	Read only	Mandatory
DHCPServer	DHCP server	DHCP server	No	No	No	No
History	History / Referenced object	History / Referenced object	Yes	No		
Location	Location	Location	No	No	No	No

The bottom section shows the configuration for the **DHCPServer** table:

- SQL Name: DHCPServer
- Label: DHCP server
- Description: DHCP server
- Foreign key: IDHCPServerId
- Target table: amComputer
- Reverse link: IPRangesManaged
- Fields: [Searchable]
- System object: No
- Domain: [Dropdown]
- Screen set: [Dropdown]
- Type: Neutral
- Size: [Dropdown]
- User type: Normal
- Format: [Dropdown]
- Upgrade: [Dropdown]

1. In the Asset Manager Application Designer, navigate to the IPRange table.
2. Create the link Location:
 - Link type: **Normal**
 - Source SQL Name: **Location**
 - Source Label: **Location**
 - Source Description: **Location**
 - Create an index: Checked
 - Destination Table: **Locations**

- Destination SQL Name: **IPRanges**
- Destination Label: **IP ranges**
- Destination Description: **IP ranges**

3. Create the link DHCP server:

- Link type: **Normal**
- Source SQL Name: **DHCPServer**
- Source Label: **DHCP server**
- Source Description: **DHCP server**
- Create an index: **Checked**
- Destination Table: **Computers**
- Destination SQL Name: **IPRangesManaged**
- Destination Label: **IP ranges managed**
- Destination Description: **IP ranges managed by this DHCP server**

Add an index to amIPRange

The screenshot shows the configuration for the **amIPRange** table and the **IPRang_EndAddress** field.

Table Configuration (amIPRange):

SQL name	Label	Description	System object
IPRang_EndAddress	End address	IP range end address	No
IPRang_IDHCPServer	ID of link DHCP server	ID of link DHCP server	No
IPRang_IPRangeld	IPRang_IPRangeld	IPRang_IPRangeld	Yes
IPRang_LocationId	ID of link Location	ID of link Location	No
IPRang_StartAddress	Start address	IP range start address	No
IPRang_Status	Status	Status	No
IPRange	IP range	IP range	No

Field Configuration (IPRang_EndAddress):

SQL Name: IPRang_EndAddress
 Label: End address
 Description: IP range end address
 System object: No
 Type: Normal
 Fields: EndAddress

1. In the Asset Manager Application Designer, navigate to the IP Ranges table.

2. Create the index IPRange:

- SQL Name: **IPRange**
- Label: **IP range**
- Description: **IP range**
- Fields: **StartAddress, EndAddress**
- Type: **Normal**

Add a validity script to amIPRange

SQL Name: amIPRange
Label: IP ranges
Description: IP ranges
Hierarchy: Can contain features

System object: No
String: [IPRangeId]
Validity: [Script]
Relevance: Yes

Basic Script

'Validity' of table 'IP ranges (amIPRange)'
Expected return type = 'Integer'

Field: Status

```
FunctionRetVal() As Integer
If [StartAddress] >= [EndAddress] then
  Err.Raise(-1, "The value of the 'Start address' field must be less than the value of the 'End address' field.")
 RetVal = FALSE
else
 RetVal = TRUE
end if
End Function
```

OK Cancel

1. In the Asset Manager Application Designer, navigate to the IP Ranges table.
2. Set the Validity field to **(Script)** and click on the magnifying glass to open the script editor.
3. Enter the following script:

```
If [StartAddress] >= [EndAddress] then
  Err.Raise(-1, "The value of the 'Start Address' field must be less
  than the value of the 'End Address' field.")
 RetVal = FALSE
else
 RetVal = TRUE
end if
```


Add a page to amIPRange

The screenshot shows the configuration for the **amIPRange** page. The **General** tab is active, and the **Contents** sub-tab is selected. The configuration includes:

- SQL Name:** amIPRange
- Label:** IP ranges
- Description:** IP ranges
- System object:** No
- String:** [IPRangeId]
- Validity:** [Script]
- Relevance:** Yes

SQL name	Label	Description	System object
amIPRange		IP range	No
PgGenDoc	Documents	Documents	Yes
PgHist	History	History	Yes

Below the table, the **Contents** sub-tab is active, showing the following configuration:

- SQL Name:** amIPRange
- System object:** No
- Domain:** Portfolio
- Screen set:** Full
- Type:** [Dropdown]
- Size:** [Dropdown]
- User type:** [Dropdown]
- Format:** [Dropdown]
- Upgrade:** [Dropdown]

1. In the Asset Manager Application Designer, navigate to the IP Ranges table.
2. Create the page amIPRange:
 - **SQL Name:** **amIPRange**
 - **Label:** empty
 - **Description:** **IP range**
 - **Domain:** **Portfolio**
3. Click the **Contents** tab and add **StartAddress**, **EndAddress**, **Status**, **Location**, and **DHCPsServer** to the list of fields.

Create a screen for amIPRange

The screenshot shows the configuration for the **IPRange** screen. The **General** tab is active, and the **Pages** sub-tab is selected. The configuration includes:

- SQL Name:** amIPRange
- Label:** IP ranges
- Description:** IP ranges
- System object:** No
- String:** [IPRangeId]
- Validity:** [Script]
- Relevance:** Yes

Screen set	SQL name	Label	Description
Full	IPRange	IP ranges	IP ranges

Below the table, the **Pages** sub-tab is active, showing the following configuration:

- SQL Name:** IPRange
- System object:** No
- Domain:** Portfolio
- Screen set:** Full
- Type:** [Dropdown]
- Size:** [Dropdown]
- User type:** [Dropdown]
- Format:** [Dropdown]
- Upgrade:** [Dropdown]

1. In the Asset Manager Application Designer, navigate to the IP Ranges table.

2. Create a new screen with the following fields:
 - SQL Name: **IPRange**
 - Domain: **Portfolio**
 - Label: **IP ranges**
 - Screen set (Asset Manager 5 and greater): **Full**
 - Description: **IP ranges**

Add a page to amLocation

The screenshot shows the configuration for a new page named PglPFRange. The main configuration area is for the screen 'amLocation' (Label: Locations, Description: Locations). Below this, a table lists existing pages:

SQL name	Label	Description	System object
PgHostCable	Host cables	Host cables	Yes
PglPFRange	IP Range	IP Range for this location	No

The 'Contents' tab is active, showing the configuration for the page 'PglPFRange':

- SQL Name: PglPFRange
- System object: No
- Domain: Portfolio
- Label: IP Range
- Description: IP Range for this location
- Screen set: (empty)
- Type: (empty)
- Size: (empty)
- User type: (empty)
- Format: (empty)
- Upgrade: (empty)

1. In the Asset Manager Application Designer, navigate to the Locations table.
2. Create the page PglPFRanges with the following fields and values:
 - SQL Name: **PglPFRanges**
 - Label: **IP ranges**
 - Description: **IP ranges**
 - Domain: **Portfolio**
3. Click the **Contents** tab and add **IPRanges** to the list of fields.

Modify a screen for amLocation

The screenshot shows the configuration for the 'Pages' tab of the screen 'amLocation' (Label: Locations, Description: Locations). The 'Available pages' list contains 'PgAcwComment'. The 'Pages in detail' list contains 'sysamLocation' and 'PglPFRange' (highlighted).

1. In the Asset Manager Application Designer, navigate to the Locations table.
2. Navigate to the screen on which you would like to add the new page.

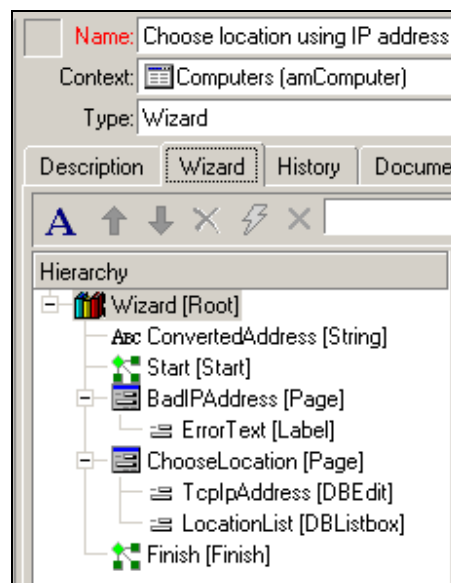
3. Click the **Pages** tab and add **PgIPRanges** to the “Pages in detail” list.
4. Save the modifications to the database.

Create a wizard

The screenshot shows a configuration dialog for a wizard. The fields are as follows:

- Name:** Choose location using IP address
- Context:** Computers (amComputer)
- Type:** Wizard
- SQL name:** Choose_location_using_IP_address
- Domain:** /Portfolio management/IT/Inventory/
- Nature:** (empty dropdown)
- Author:** Admin
- Visible in the menu:**

1. In the Windows® client, navigate to the Actions screen.
2. Click **New** and enter the description for the new wizard, as follows:
 - Name: **Choose location using IP address**
 - Context: **Computers (amComputer)**
 - Type: **Wizard**
 - SQL name: **Choose_location_using_IP_address**
 - Domain: **/Portfolio management/IT/Inventory/**
 - Visible in the menu: **Checked**



3. Click the **Wizard** tab, and click the Graphical/Text editor button to switch to text view.

4. Enter or paste the following script:

```
Name = "Wizard"
Version = "5479"
{ String ConvertedAddress
  { Value =
    on error resume next

    dim seg1 as string
    dim seg2 as string
    dim seg3 as string
    dim seg4 as string
    dim CompIPValue as string
    dim CompNewIPValue as string
    dim lErr as long

    CompIPValue = [TcpIpAddress]

    seg1= ExtractValue( CompIPValue , ".", "\")
    seg2= ExtractValue( CompIPValue , ".", "\")
    seg3= ExtractValue( CompIPValue , ".", "\")
    seg4= ExtractValue( CompIPValue , ".", "\")

    if Len(seg2) < 3 then
      if Len(seg2) < 2 then
        CompNewIPValue = seg1 & "00" & seg2
      else
        CompNewIPValue = seg1 & "0" & seg2
      end if
    else
      CompNewIPValue = seg1 & seg2
    end if

    if Len(seg3) < 3 then
      if Len(seg3) < 2 then
        CompNewIPValue = CompNewIPValue & "00" & seg3
      else
        CompNewIPValue = CompNewIPValue & "0" & seg3
      end if
    else
      CompNewIPValue = CompNewIPValue & seg3
    end if

    if Len(seg4) < 3 then
      if Len(seg4) < 2 then
        CompNewIPValue = CompNewIPValue & "00" & seg4
      else
        CompNewIPValue = CompNewIPValue & "0" & seg4
      end if
    else
      CompNewIPValue = CompNewIPValue & seg4
    end if

   RetVal = CompNewIPValue
  }
}

{ Start Start
  { To =
    If isNumeric({ConvertedAddress}) then
      RetVal = "ChooseLocation"
    else
      RetVal = "BadIPAddress"
    end if
  }
}

{ Page BadIPAddress
  OnEnter = ""
}
```

```

Title = "Error"
{ Label ErrorText
  Caption = "The wizard can not understand the IP address for this
  device. Please use 4 sections with three numbers each separated
  with periods like: 128.001.001.001"
}
}

{ Page ChooseLocation
{ DBEdit TcpIpAddress
  Enabled = 0
  Field = "TcpIpAddress"
  Label = "This computer's IP address"
  ReadOnly = 1
  Table = "amComputer"
  Value = [TcpIpAddress]
}
}
ListBox LocationList
ColTitle = "Location|Start Address|End Address"
ColWidth = "50|15|15"
Label = "Possible locations for this computer"
Mandatory = 1
Sortable = 1
Value = "-1"
{ Values =
  dim seg1 as string
  dim seg2 as string
  dim seg3 as string
  dim seg4 as string
  dim CompIPValue as string
  dim CompNewIPValue as string
  dim lErr as long
  dim where as string

  CompIPValue = [TcpIpAddress]

  seg1= ExtractValue( CompIPValue , ".", "\")
  seg2= ExtractValue( CompIPValue , ".", "\")
  seg3= ExtractValue( CompIPValue , ".", "\")
  seg4= ExtractValue( CompIPValue , ".", "\")

  if Len(seg2) < 3 then
    if Len(seg2) < 2 then
      CompNewIPValue = seg1 & "00" & seg2
    else
      CompNewIPValue = seg1 & "0" & seg2
    end if
  else
    CompNewIPValue = seg1 & seg2
  end if

  if Len(seg3) < 3 then
    if Len(seg3) < 2 then
      CompNewIPValue = CompNewIPValue & "00" & seg3
    else
      CompNewIPValue = CompNewIPValue & "0" & seg3
    end if
  else
    CompNewIPValue = CompNewIPValue & seg3
  end if

  if Len(seg4) < 3 then
    if Len(seg4) < 2 then
      CompNewIPValue = CompNewIPValue & "00" & seg4
    else
      CompNewIPValue = CompNewIPValue & "0" & seg4
    end if
  else
    CompNewIPValue = CompNewIPValue & seg4
  end if
}
}

```


3. Click the **Script** tab and enter or paste the following script:

```
'Error codes
Const AME NODATA = 12004

dim lErr as long

'Grab all of the computers that need to processed
dim Qry As String
Qry = ""
Qry = Qry & "SELECT lComputerId, TCPIpAddress, Portfolio.lLocaId "
Qry = Qry & " FROM amComputer "
Qry = Qry & " WHERE Portfolio.lLocaId = 0 AND TCPIpAddress <> '' "

dim hqComputers As Long
dim lComputersErr as long
lComputersErr = 0
hqComputers= AmQueryCreate()
lComputersErr = AmQueryExec(hqComputers, Qry)

'For each computer...
Do While lComputersErr <> AME_NODATA

    dim lLocaId As Long
    dim seg1 as string
    dim seg2 as string
    dim seg3 as string
    dim seg4 as string
    dim CompIPValue as string
    dim CompNewIPValue as string

    CompIPValue = AmGetFieldStrValue(hqComputers,1)

    '...convert the IP address to a number...
    seg1= ExtractValue( CompIPValue , ".", "\")
    seg2= ExtractValue( CompIPValue , ".", "\")
    seg3= ExtractValue( CompIPValue , ".", "\")
    seg4= ExtractValue( CompIPValue , ".", "\")

    if Len(seg2) < 3 then
        if Len(seg2) < 2 then
            CompNewIPValue = seg1 & "00" & seg2
        else
            CompNewIPValue = seg1 & "0" & seg2
        end if
    else
        CompNewIPValue = seg1 & seg2
    end if

    if Len(seg3) < 3 then
        if Len(seg3) < 2 then
            CompNewIPValue = CompNewIPValue & "00" & seg3
        else
            CompNewIPValue = CompNewIPValue & "0" & seg3
        end if
    else
        CompNewIPValue = CompNewIPValue & seg3
    end if

    if Len(seg4) < 3 then
        if Len(seg4) < 2 then
            CompNewIPValue = CompNewIPValue & "00" & seg4
        else
            CompNewIPValue = CompNewIPValue & "0" & seg4
        end if
    else
        CompNewIPValue = CompNewIPValue & seg4
    end if

end if
```

```

If isnumeric(CompNewIPValue) then

    '...check for possible location matches
    Qry = ""
    Qry = Qry & "SELECT lLocaId "
    Qry = Qry & " FROM amLocation"
    Qry = Qry & " WHERE (IPRanges.StartAddress <> 0 AND
    IPRanges.EndAddress <> 0) AND (IPRanges.StartAddress <=" &
    CompNewIPValue & ") AND (IPRanges.EndAddress>=" & CompNewIPValue &
    ")"
    Qry = Qry & " ORDER BY slvl DESC"

    dim hqLocations As Long
    dim lLocationsErr as long
    lLocationsErr = 0
    hqLocations = AmQueryCreate()
    lLocationsErr = AmQueryExec(hqLocations, Qry)

    If lLocationsErr <> AME NODATA then
        lLocaId = AmGetFieldLongValue(hqLocations,0)

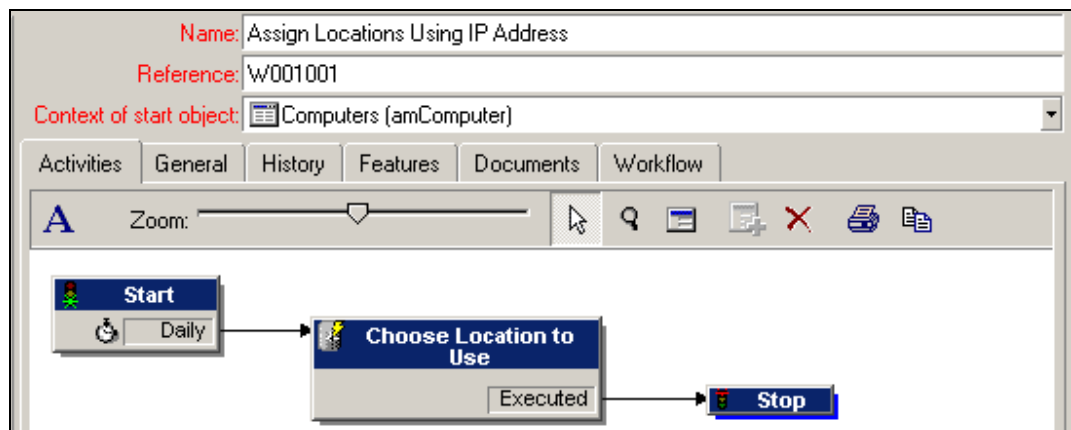
        '...apply the first match to the computer
        lErr = AmStartTransaction()
        dim hrComputer as long
        hrComputer = AmGetRecordHandle(hqComputers)
        lErr = amSetFieldLongValue(hrComputer, "Portfolio.lLocaId",
        lLocaId)
        lErr = amUpdateRecord(hrComputer)
        lErr = AmCommit()
    end if
end if

' Get next Work Order request line
lComputersErr = AmQueryNext(hqComputers)

Loop
RetVal = 0
Exit Function

```

Create the workflow



1. In the Windows client, navigate to the Workflow schemes screen.
2. Click **New** and enter the description for the new workflow as follows:
 - Name: **Assign locations using IP address**
 - Context of start object: **Computers (amComputer)**

3. Click **Create**.
4. For the Start activity, double click the empty event.
5. Specify the parameters for the event, as follows:
 - Name: **Start**
 - Type: **Periodical**
6. Click the **Parameters** tab, specify the desired schedule, and click **Add**.
7. Create a new activity and enter its description, as follows:
 - Name: **Choose location to use**
 - Type: **Automatic action**

The screenshot shows a configuration window for an action. The 'Name' field contains 'Choose Location to Use' and the 'Type' dropdown is set to 'Automatic action'. Below these fields are several tabs: 'General', 'Parameters' (which is selected), 'Time limit', 'Alarms', 'History', and 'Documents'. Under the 'Parameters' tab, there is a checked checkbox for 'Execute actions immediately'. At the bottom, there is a table with two columns: 'Name' and 'Type'. The table contains one row with the value 'Assign location using IP address' under 'Name' and 'Script' under 'Type'.

Name	Type
Assign location using IP address	Script

8. To the list of actions, add **Assign location using IP address** and click **Add**.
9. Create a transition between the start event and the Choose location activity.
10. Right-click the executed event and click **Terminal Event**.

Limitations of the solution

When implementing and using this solution, keep in mind that the scripts assume that locations that are deeper in the locations tree will be a better match; and that the scripts will present them higher in the list of possible matches. This could affect the accuracy of the script, especially in the workflow. At the same time, this could increase accuracy in cases where the IP ranges are assigned as a tree (with the parent location having a broader range than its children).

For example, if we have an arrangement such as the following:

- Building A: 15.0.0.0 to 15.0.1.0
- Floor 1: 15.0.0.1 to 15.0.0.100

If you are trying to find the location for 15.0.0.50, the script will present Floor 1 before Building A. Both ranges are a possible match, but since Floor 1 is deeper in the locations tree, the script assumes that it is more specific and is a better match.

If you are trying to find the location for 15.0.0.200 and there are no floors with this range, the broader range of Building A would be recommended as a possible match.

Ways to extend this solution

While this solution provides the functionality to automate location assignment, some organizations may have additional requirements. Here are some ideas on how this solution could be extended:

- Add a button to the Computers table that triggers the wizard.
- Add the location link to the Computers screen.

- Add additional logic to the workflow to help it determine the best match rather than taking the first location match.
- Use the status field to restrict the matches that are presented in the wizard and workflow.

Summary

This solution is meant to help organizations automate a very challenging part of the asset management process—maintaining accurate asset location data. If the organization has a discovery tool that supplies IP addresses for network devices, Asset Manager can correlate that information, with IP ranges associated with physical locations. Rather than being forced to perform a physical inventory, the organization can use IP addresses to fill in some of the missing asset data. By extending the Asset Manager data model with an IP Ranges table, and by adding a wizard and a workflow, your organization can have a powerful new tool to save time and improve data accuracy.

For more information

Please visit the HP Software support Web site at:

<http://www.hp.com/managementsoftware/support>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Note: Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to the following URL:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to the following URL:

<http://www.managementsoftware.hp.com/passport-registration.html>

Limited responsibility clause

Asset Manager is integrated with several third-party applications. Examples: Database engines, Web servers, single sign-on software, load-balancing and clustering hardware and software solutions, reporting software such as Crystal Reports, etc.

Support for these applications is limited to their interface with Asset Manager. Support does not cover installation problems, setup and customization problems nor malfunctioning of the third-party application.

White papers contain examples of implementations that may work in your environment with or without customization. There is no guarantee that this will be the case. It could also be that some of the solutions covered by white papers appear as standard features in a future release of the software. When this is the case, there is no guarantee that you will be able to upgrade the solution you implemented based on the white paper to the equivalent standard feature.

© 2008 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Itanium is a trademark or registered trademark of Intel Corporation in the U.S. and other countries and is used under license.

Using IP addresses to set asset location in AM.doc



invent