

Best Practices for Multi-Tenant Environments

How to Configure Service Manager 7 for Multi-Tenant Implementations.

HP® Software — Service Management



Introduction	2
Techniques used in Multi-Tenant implementations	2
Data Segregation	2
Mandanten security	2
Data Presentation	2
Dynamic forms	2
Dynamic sub-formats	2
Pre-requisites	4
Tailoring	4
Setting up Data Segregation	5
Plan your Data Segregation strategy	5
Setting up the Mandanten values	5
Customizing Service Manager for setting the company context	6
Create company specific subformats	12
Invoking the company context when opening an incident	13
Creating dynamic sub-formats	16
Setting the system up for dynamic sub-formats	16
Creating the dynamic subform from the company record	18
Appendix A	20
Checklist of records to modify to allow for this functionality	20
Service and Incident Management	20
Additional steps to perform on Change Management	21
For more information	22

Introduction

From the Service Desk perspective the Service Manager application is designed primarily to provide basic support within an organization. Managed Service Providers (MSP) or Software as a Service (SaaS) who provide Service Desk support to multiple tenants, or large diverse organizations that provide support to multiple clients will need to extend Service Manager functionality. Extending Service Manager functionality to accommodate these complex organizations with multiple tenants requires several modifications to the data presentation and data segregation methodologies.

All steps described in this document should be performed by an experienced system administrator.

Techniques used in Multi-Tenant implementations

Data Segregation

Data Segregation is necessary in the multi-tenant environment to ensure that one tenant cannot access data from other tenants. The implementation of Mandanten security to segregate data in Service Manager is needed to perform the segregation.

Mandanten security

Mandanten protection should be implemented against tables such as *probsummary*, *company*, *contacts*, *category*, *subcategory*, *producttype* and *problemtype*, as well as other supporting tables such as *device* and *department*. The number of tables that should be protected depends on the environment and the modules implemented.

For additional information about Mandanten table security and the `mandant()` function, see the Service Manager online help.

Using the `mandant()` function to implement data segregation

As an extension to the data segregation, we will implement the capability for users who support multiple tenants to be able to choose the tenant environment to work in. Use the `mandant()` function – the correct syntax is: `mandant(2,{array_of_security_groups})` – to set and reset Mandanten values to show either records for all of the user's security groups, or to show only the security group that is specific to the customer calling in.

When the help desk employee selects the company (or department) from which the customer is calling, the `mandant()` function sets the Mandanten environment (all Mandanten protected tables) to display only records that are relevant to that company. After the employee finishes supporting a specific tenant and returns to his or her work queue, tickets for all of the employee's security groups are displayed once again.

Data Presentation

Dynamic forms

In Service Manager, form names can be dynamically changed using the Document Engine tailoring tool. For example, if an employee is allowed to view some records and update others, or is allowed only a restricted view of the data, it is more efficient to change the form used to display the data or the I/O condition with a single expression in the Document Engine than by using DVD statements on the form to manipulate the display properties of each element on the form.

Dynamic sub-formats

Service Manager provides sub-formats to display specialized information. By default, the name of a sub-format is hard-coded using Forms Designer so that the sub-format is static. A dynamic sub-format

is a sub-format linked to the main form where a variable is used to specify the name of the sub-format so that it can be loaded dynamically at runtime.

With dynamic sub-formats, generic information such as ticket categorization, contact information, Service Level Agreement (SLA) information, ticket history, and ticket alerting are put into the main form such as IM.template.update. All tenant-specific information can be put onto a single sub-format. As the helpdesk user who supports multiple tenants' scrolls from ticket to ticket, the sub-format display dynamically changes to the unique information and format required for each tenant.

To reduce the effort of creating these company-specific sub-formats, HP Software created an application to generate them automatically, based on information in the company table. The administrator simply enters the labels, field names and required field types as well as a limited amount of DVD (Dynamic View Dependency) values into the company record and saves the record. A sub-format with the correct naming convention (company.<company_name>.<tablename>.subformat) based on the requested information is automatically generated.

Warning: If a sub-format with that name already exists, it is overwritten.

Pre-requisites

1. Each Security Group must represent exactly one company.
2. Security Group names must be in upper-case.
3. The Company field must be required in Service Desk interactions as well as all other tables that need to be Mandanten protected.
4. Application Threading must be completely turned off for the user in each module where this is applied. Typically, this is specified in the Profile records for the various modules in the System (Service Desk, Incident Management, Configuration Management, Change Management, etc.).

Several unload files accompany this document in a zip file called *MSP_Best_PracticesSM7.zip*.

Note: These files were created using Service Manager 7, and should be applied only to Service Manager 7 systems.

Follow the instructions in the tailoring and dynamic sub-format sections for loading the unload files.

Important: Unload files are clearly marked *OOBonly* if they should be applied only to an Out-Of-Box system. Unload files that do not include *OOBonly* in their name can be applied to any system.

Tailoring

The following steps show how to tailor the Incident Management module. All tailoring is described in detail, but can be made easier by loading the supplied unloads. A file named *SM7TailoringOOBonly.unl* is included in the zip supplied with this document. This file should only be loaded into an out-of-box system and will load the following records:

Table Name	Record
operator	BOB.HELPDESK
scmandant	contacts, dept, device, probsummary
scsecuritygroup	ACME, Company A, Company B, Company C, GENERICOM, HP, TEST
formatctrl	login.DEFAULT
dbdict	probsummary
format	IM.template.browse.g, IM.template.close.g, IM.template.km.g, IM.template.open.g, IM.template.ro.g, IM.template.update.g, apm.quick.g
displayscreen	scm.advanced
Object	probsummary
Process	cc.create, cc.qopen.create, us.screlate, im.view.init, im.new.incident
link	incidents, probsummary
menu	IM

Additionally you can load the *SM7TailoringCustom.unl* file into any system. This file will load the following records:

Table Name	Record
Process	company.specific.incident, company.specific.incident2
Format	company.prompt.g
Wizard	MSP Reset Company, MSP Switch Company, MSP Switch Company 2

Setting up Data Segregation

Plan your Data Segregation strategy

Determine which tables need to be protected and who will have access to which records. This includes making sure that the company field is part of each of these tables.

Setting up the Mandanten values

1. The scmandant table should have records for the probsummary and company tables and for all other tables in your setup that need to be protected based on the value of the company field.

Mandanten Field Restriction

File Name:

Mandant Field Name:

Linkage Field Name:

Source File Name:

Source Field Name:

Exclude field:

2. Create the scsecuritygroup records to match the companies (see prerequisites 1 and 2)

Mandanten Security Groups

Security ID:

Include Value List:

Exclude Value List:

3. Modify the operator records to include all appropriate security groups.

Operator Record

Security Groups:

Customizing Service Manager for setting the company context

1. Add a calculation to the login.DEFAULT FormatControl to determine the security groups an operator has upon login. The add condition for the calculation is TRUE, and the expression is:

\$G.orig.secgroup=security.group in \$file

2. Create a Process named "company.specific.incident" that sets the Mandanten values based on the company selected; and then calls the apm.first application:

On the initial expressions tab enter:

\$L.security.token=str(toupper(company in \$L.file))

\$L.void=mandant(2, {\$L.security.token})

On the RAD tab, enter in the area labeled "Expressions evaluated before RAD call"

\$G.company.sub.format="company."+tolower(company in \$L.file)+"."+filename(\$L.file)+".subformat"

And then call **apm.first** with a condition of **not null(company in \$L.file)** as shown below:

Process Definition

Process Name:

Save Cursor Position? Run Standard Process when complete?

Run in Window? Window Title:

◆ Initial Expressions ◆ Initial Javascript ◆ **RAD** ◆ Final Expressions ◆ Final Javascript ◆ Next Process

Expressions evaluated before RAD call

RAD Application: Condition:

Parameter Names	Parameter Values

Post RAD Expressions

3. Create a Process called company.specific.incident2 Process that sets only the Mandanten values.

On the initial expressions tab enter:

\$L.security.token=str(toupper(company in \$L.file))

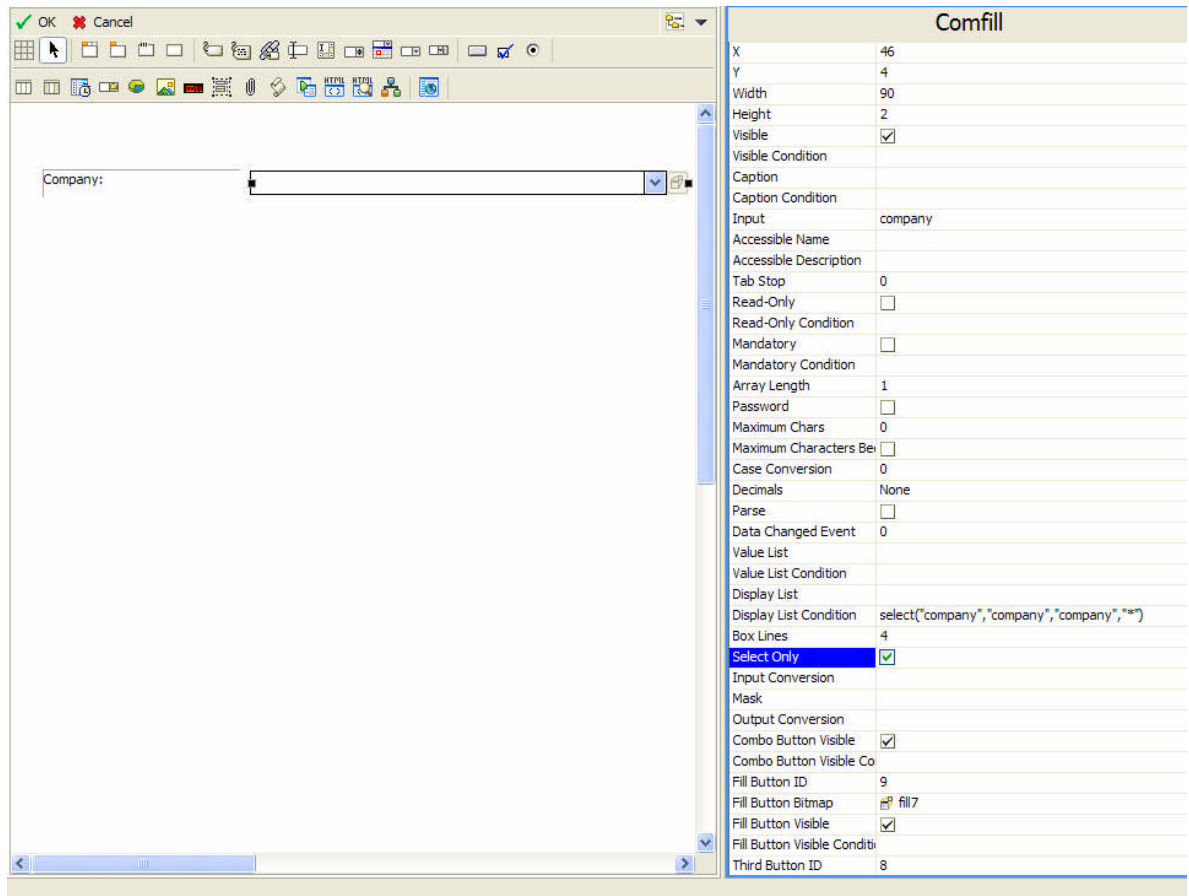
\$L.void=mandant(2, {\$L.security.token})

On the RAD tab, enter in the area labeled "Expressions evaluated before RAD call"

\$G.company.sub.format="company."+tolower(company in \$L.file)+"."+filename(\$L.file)+".subformat"

4. Create a form named **company.prompt.g** that enables selection of the company in the wizard.

Note: Make sure to check the **Select Only** property of the **Company** drop-down list shown below



5. Create three wizards. These are used to execute the mandant () function to restore the default login Mandanten settings for the operator, and reset the current company setting.

- **MSP Switch Company**

On the Wizard Info tab, enter a text such as **“Select Company to create Incident for”** for Window Title, Title, and prompt.

Wizard Information	
Wizard Name:	MSP Switch Company <input type="checkbox"/> Start Node?
Brief Description:	Dynamically switch company context for MSP cust
Window Title:	Select Company to create Incident for
Title:	Select Company to create Incident for
Prompt:	Select Company to create Incident for
Bitmap:	
Global Lists:	

On the File Selection tab, enter the Initial Expression **\$L.void=mandant(2, \$G.orig.secgroup)** and select the \$L.file by creating a **probsummary** type record.

Wizard Information

- Wizard Info
- File Selection
- Usage
- Actions
- Mess:

- Initial Expressions
- Javascript
- Select \$.file by

```
$L.void=mandant(2, $G.orig.secgroup)
```

Wizard Information

- Wizard Info
- File Selection
- Usage
- Actions
- Messaging
- Variables
- Next Wizard
- Comments
- Cancel Expressions

- Initial Expressions
- Javascript
- Select \$.file by

No \$.file (use typecheck)
 \$.file passed in
 Create a record of type:
 Select records of type:
 using query:
 Resolve Variables
 No Records Message:

On the Usage tab, request user input in the **company.prompt.g** form.

Wizard Information

- Wizard Info
- File Selection
- Usage
- Actions

Wizard Usage

- Select one record from list
- Request user input
- Skip Display
- Cancel Immediately
- Select multiple records from list

Sub Format to Display:

Display Screen:

(Defaults to wizard.display)

- Activate "Finish" option?
- Turn off Next and Previous buttons?
- Turn off Next button only?
- Turn off Previous button only?

On the Actions tab, select to perform all actions on the **current file** and restart the panel if **null(company in \$.file)** then call the **company.specific.incident** Process.

Wizard Information

Wizard Info | File Selection | Usage | **Actions** | Messaging | Variables | Next Wizard | Comments | Cancel Expressions

Initial Process:

Perform Actions On:

- Current File (\$.file)
- Selection (\$.selection)
- Each record in selection (\$.selection)

Actions to Perform:

Expressions | Javascript | **Format Control / Process Name**

Format Control:

of type:

on bad validation:

Process Name:

Reset Current File to Selections?

Restart Panel if:

Display Record(s) when complete? Mode:

Return Current File to calling application?

On the Next Wizard tab, call the **MSP Reset Company** wizard under the condition of **not null(company in \$.file)**.

Wizard Information

Wizard Info | File Selection | Usage | Actions | Messaging | Variables | **Next Wizard** | Comments | Cancel Expressions

Next Wizard:

Wizard Name	Condition
MSP Reset Company	not null(company in \$.file)

• **MSP Reset Company**

On the File Selection tab enter the Initial Expression **\$.void=mandant(2, \$G.orig.secgroup)**.

Wizard Information

Wizard Info | **File Selection** | Usage | Actions | Mess

Initial Expressions | Javascript | Select \$.file by

On the Usage tab, select **Skip Display**.



Wizard Information

Wizard Info
 File Selection
 Usage
 Actions
 M



Wizard Usage

Select one record from list
 Request user input
 Skip Display
 Cancel Immediately
 Select multiple records from list

Sub Format to Display:

Display Screen:

(Defaults to wizard.display)

Activate "Finish" option?
 Turn off Next and Previous buttons?
 Turn off Next button only?
 Turn off Previous button only?

- **MSP Switch Company 2** (which is copy of MSP Switch Company without the call to the company-specific incident Process)

On the Wizard Info tab, enter a text such as **“Select Company to create Incident for”** for Window Title, Title, and prompt.

Wizard Information

Wizard Info
 File Selection
 Usage
 Actions
 Messaging
 Variables
 Next W

Wizard Name:

Brief Description:

Window Title:

Title:

Prompt:

Bitmap:

Global Lists:

On the File Selection tab, enter the Initial Expression **\$L.void=mandant(2, \$G.orig.secgroup)** and select the \$L.file by creating a **probsummary** type record.

Wizard Information

Wizard Info
 File Selection
 Usage
 Actions
 Mess:

Initial Expressions
 Javascript
 Select \$L.file by

\$L.void=mandant(2, \$G.orig.secgroup)

On the Usage tab, request user input in the **company.prompt.g** form.



Wizard Information

Wizard Info
 File Selection
 Usage
 Actions



Wizard Usage

Select one record from list
 Request user input
 Skip Display
 Cancel Immediately
 Select multiple records from list

Sub Format to Display:

company.prompt.g  

Display Screen:

(Defaults to wizard.display)

Activate "Finish" option?
 Turn off Next and Previous buttons?
 Turn off Next button only?
 Turn off Previous button only?

On the Actions tab, select to perform all actions on the **current file** and restart the panel if **null(company in \$L.file)** then call the **company.specific.incident2** Process.

Wizard Information

Wizard Info | File Selection | Usage | **Actions** | Messaging | Variables | Next Wizard | Comments | Cancel Expressions

Initial Process:

Perform Actions On:

- Current File (\$.file)
- Selection (\$.selection)
- Each record in selection (\$.selection)

Actions to Perform:

Expressions | Javascript | **Format Control / Process Name**

Format Control:

of type:

on bad validation:

Process Name:

Reset Current File to Selections?

Restart Panel if:

Display Record(s) when complete? Mode:

Return Current File to calling application?

Create company specific subformats

1. Add company-specific fields to the probsummary dbdict (and if paging is used to the problem dbdict). Add some character, some number, some date and some array fields. Give them generic names such as company.text.1, company.number.1, company.date.1 etc.

Note: Modify the company.g form to have a drop down list of these available field names available for the end users to prevent confusion or missing functionality. The provided unload names these fields specific.field.1 etc.

2. Manually create a company-specific sub-format as pictured below for each of the company records in your system using the following naming convention:

company.<company_name_in_lower_case>.<tablename>.subformat.

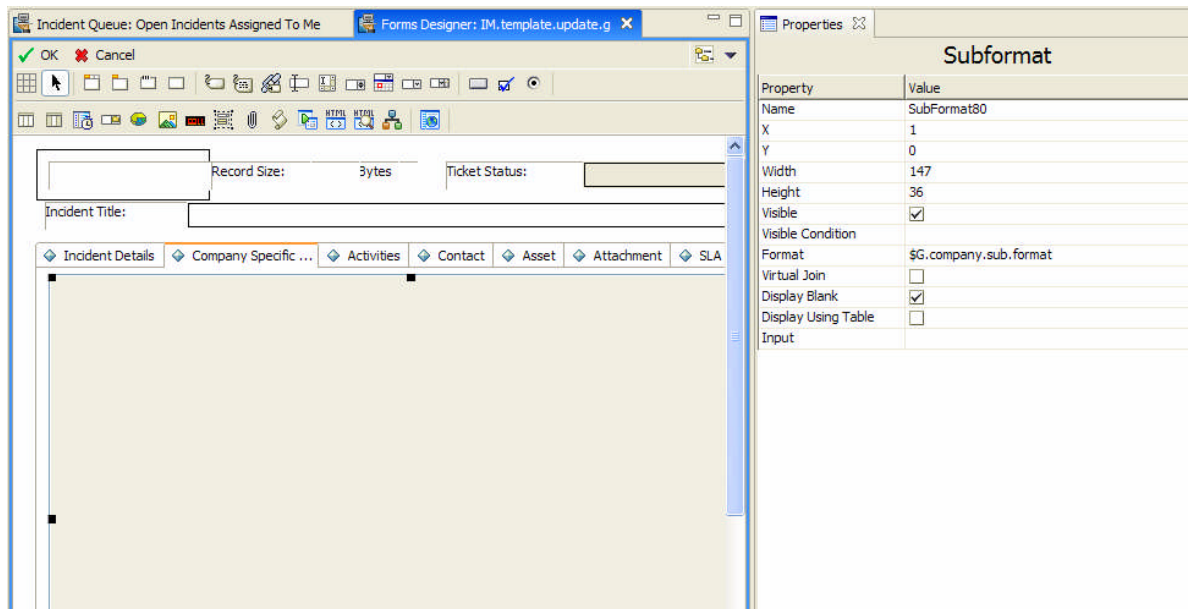
As an alternative, you can use form settings from the company table and the subform.create application that HP Software provides to create sub-formats dynamically. See the section Creating dynamic sub-formats on page 16 for more information.



ACME

Contact at Site	<input type="text"/>
System ID	<input type="text"/>
Site Manager	<input type="text"/>
License #	<input type="text"/>
Location #	<input type="text"/>

3. Modify the apm.quick and IM.template.* forms to include a tab that displays the company-specific sub-format. This is done by:
 - Adding a company-specific tab to the existing notebook Objects on these forms; or reusing an existing tab.
 - Adding a sub-format Object to the company-specific tab. For the Format property of the sub-format Object, specify the global variable `$G.company.sub.format`.



Invoking the company context when opening an incident

1. Modify the scm.advanced display screen to fill the name of the MSP Switch Company wizard into a variable called \$wizard.name. Enter in Initialization Expressions: **\$wizard.name="MSP Switch Company"**
2. Modify the probsummary Object record to call the wizard.run application as the Add/ Open Application, which calls the wizard with the name parameter = \$wizard.name).

Object Definition

File name: Unique key:

Common name:

Locking
 Revisions
 Variables/Global ...
 Activities
 Alerts
 Approvals
 Manage Queues
 Views/Templates >>

Manage condition:
Manage display format:
Manage default view:
Manage default query: Default query description:
Thread view -> search?: Search format (if necessary):
Thread search -> list?: Thread list -> edit?:
Thread view -> edit?:
Allow add condition:
Add/open application:
Parameters:

Names	Values
name	\$wizard.name

3. Modify the **cc.qopen.create** Process to set the Mandanten value based on the setting in the call, and set the correct sub-format:

if (not null(company in \$L.file)) then (\$L.void=mandant(2, {company in \$L.file});\$G.company.sub.format="company."+tolower(company in \$L.file)+"."+filename(\$L.file)+".subformat")

4. Modify the **cc.create** Process to set the Mandanten value based on the setting in the call, and set the correct sub-format:

if (not null(company in \$L.file)) then (\$L.void=mandant(2, {company in \$L.file});\$G.company.sub.format="company."+tolower(company in \$L.file)+". "+filename(\$L.file)+".subformat")

5. Add the following Initial Expressions in the **us.screlate** Process to reset the Mandanten value to base settings.

\$L.void=mandant(2, \$G.orig.secgroup)

6. Modify the **im.view.init** Process to set the sub-format to the company-specific sub-format. Add the following line in the first available line in Initialization Expressions:

\$G.company.sub.format="company."+tolower(company in \$L.file)+". "+filename(\$L.file)+".subformat"

7. Modify the **im.new.incident** Process to call the MSP Switch Company 2 wizard. (This Process is called from the im.search State.). In the Initial Expressions add:

\$L.wizard2call="MSP Switch Company 2"

On the RAD tab, call **wizard.run** with the **name** parameter value of **\$L.wizard2call** and the **text** parameter of **\$L.wiz.exit** as shown below:

Process Definition

Process Name:

Save Cursor Position? Run Standard Process when complete?

Run in Window?

Window Title:

Initial Expressions | Initial Javascript | **RAD** | Final Expressions | Final Javascript | Next Process

Expressions evaluated before RAD call

RAD Application: Condition:

Parameter Names	Parameter Values
name	\$L.wizard2call
text	\$L.wiz.exit

Post RAD Expressions

Enter the Post RAD expression of **if (\$L.wiz.exit="cancel") then (\$L.exit="exit") else (\$L.exit="new")** to correctly handle a user cancelling out of the wizard.

8. Modify the incidents link to allow the display only of categories for the contact's company. Add the following Post Link Expressions to the contact.name line:

\$L.security.token=str(toupper(company in \$File)); \$L.void=mandant(2, {\$L.security.token})

9. **(Optional)** To limit the probsummary list to display only records with the categories in your Mandanten list, add a build list subroutine to the FormatControl login.DEFAULT to build a list \$G.my.categories that contains all categories this user is allowed to see. Then, create an saccess record for probsummary with the following query:

category isin \$G.my.categories.

Note: This step is not necessary in a system where Mandanten protection was set from inception, because users would not have tickets in their queues with categories that are not allowed.

10. Change the **IM** menu to call **wizard.run** rather than apm.first, as outlined in the red space highlighted in the following screen:

Menu

Menu Name:

IM

Format:

menu.gui.pm

Opt...	Group	Description	Com...	Application	Parameter...	Parameter Value	Thr...	Condition
1		Incident Queue		sc.setup.m...	string1	problem		loption("Incident Management") and (index("Incident management", \$lo.uca...
2		Search Incidents		apm.searc...			true	index("IncidentAdmin", \$lo.ucapex)>0 or index("SysAdmin", \$lo.ucapex)>0 ...
3			Back	menu.man...	name	\$lo.home	true	name in \$mm.menu~=\$lo.home
4	Tools	Tools::Probable ...		database	name	probable.cause	true	loption("Incident Management") and (index("Incident management", \$lo.uca...
11	Tools	Tools::Summary ...		se.search....	{"name", "q...	{"link", "name="\bu...	true	index("Incident management", \$lo.ucapex)>0 or index("IncidentAdmin", \$lo...
12	Tools	Tools::Reset Do...		pm.clear.d...			true	index("SysAdmin", \$lo.ucapex)>0 or index("IncidentAdmin", \$lo.ucapex)>0
16		Search Knowledg...		get.search...			true	index("SysAdmin", \$lo.ucapex)>0 or index("IncidentAdmin", \$lo.ucapex)>0
17	Tools	Tools::Downtime		database	name	downtime.graph	true	index("Incident management", \$lo.ucapex)>0 or index("IncidentAdmin", \$lo....
22				menu.man...	name	\$lo.home	true	name in \$mm.menu~=\$lo.home
23		System Status		system.sta...			true	index("SysAdmin", \$lo.ucapex)>0
24		ServiceManager ...		mail.scenter			true	true
30		Open New Incident		wizard.run	name	MSP Switch Company	true	index("Incident management", \$lo.ucapex)>0 and evaluate(open in \$G.pm....
43	Tools	Tools::Product T...		database	name	product.type	true	index("IncidentAdmin", \$lo.ucapex)>0 or index("SysAdmin", \$lo.ucapex)>0 ...
44	Tools	Tools::Problem T...		database	name	problem.type	true	index("IncidentAdmin", \$lo.ucapex)>0 or index("SysAdmin", \$lo.ucapex)>0 ...
45	Tools	Tools::Categories		se.search....	name	category	true	index("IncidentAdmin", \$lo.ucapex)>0 or index("SysAdmin", \$lo.ucapex)>0 ...
46	Tools	Tools::Subcateg...		database	name	subcategory	true	index("IncidentAdmin", \$lo.ucapex)>0 or index("SysAdmin", \$lo.ucapex)>0 ...

Creating dynamic sub-formats

Setting the system up for dynamic sub-formats

The subform.create and us.form.builder.advanced applications will be provided with this paper in the subform.create.Apps.unl unload file. The company dbdict, form and FormatControl are supplied in the SM7DynFormOOBonly.unl unload file. This unload file can only be applied to an out-of-box system. For customized systems, follow the steps outlined below.

Content of subform.create.Apps.unl:

RAD	us.form.builder.advanced, subform.create
-----	--

Content of SM7DynFormOOBonly.unl:

dbdict	company
format	company.g
formatctrl	company

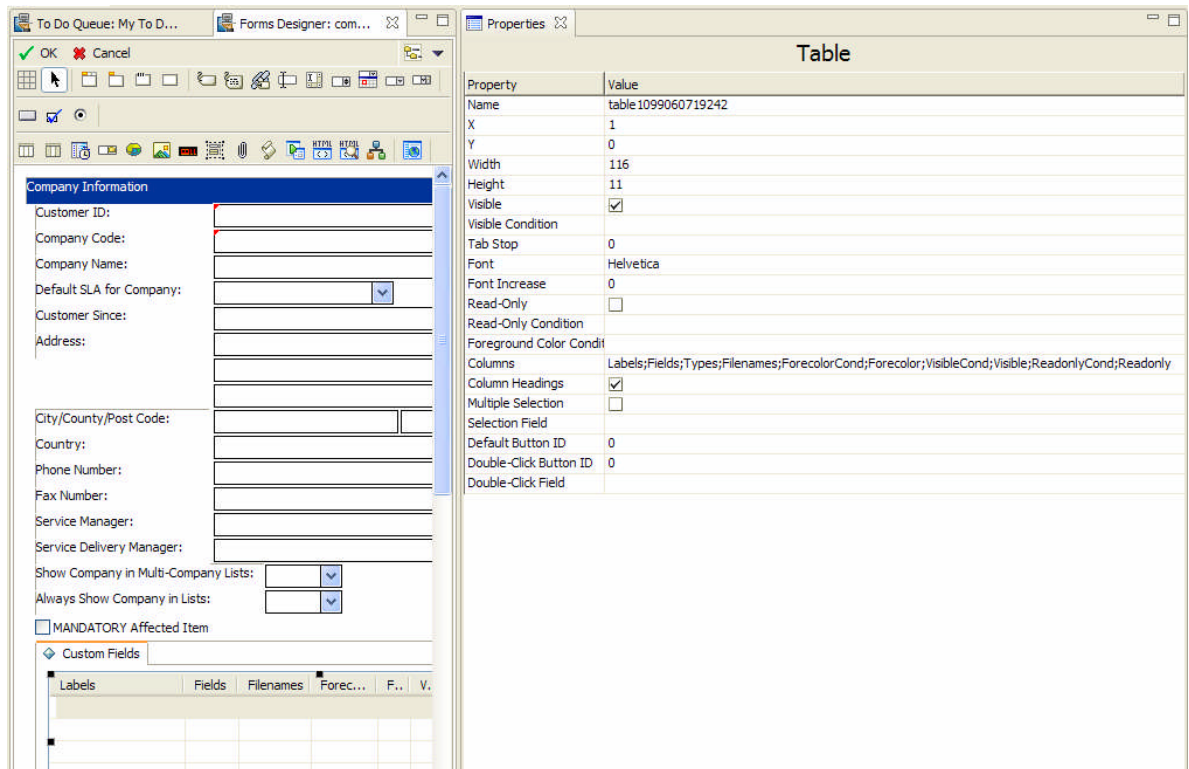
The subform.create application can create sub-forms dynamically based on information in the custom.fields structured array in the company table. Follow these steps to create a form dynamically:

1. Load the subform.create.Apps.unl unload.
2. Add the following fields to the company dbdict, first add the custom.fields array of structure, then add the other fields to the custom.fields structure

Note: The order of the fields within the structure is important and has to be kept exactly as shown below.

Field	Datatype
custom.fields	Array
custom.fields	Structure
labels	Character
fields	Character
types	Character
filename	Character
ForeColorCond	Character
ForeColor	Number
VisibleCond	Character
Visible	Number
ReadOnlyCond	Character
ReadOnly	Number

3. Add a new table to the company.g format with the following settings:



- Column Names: **Labels, Fields, Types, Filenames, ForeColorCond, ForeColor, VisibleCond, Visible, ReadOnlyCond, ReadOnly**
4. Adjust the settings of the columns within the table as follows
 - Input: **custom.fields** (same for all columns)
 - Field: Enter the field name from the table above that matches the column label.
 5. In the company FormatControl record, add the following calculations all with the condition on add / update: **true**

```

$company=company in $file
$filename.array={}
for $ind.i = 1 to lng(denull(custom.fields in $file)) do (if (index(4 in $ind.i in custom.fields in $file, $filename.array)=0) then ($filename.array=insert($filename.array, 1, 1, 4 in $ind.i in custom.fields in $file)))

```
 6. In the company FormatControl record, add the following subroutine call:

Format Control Maintenance - Subroutines

Name: company

View: long

Subroutines

Application Name	subform.create															
Comments																
Names	<table border="1"> <tr><td>name</td><td>Values</td><td>\$company</td></tr> <tr><td>file</td><td></td><td>\$file</td></tr> <tr><td>names</td><td></td><td>\$filename.array</td></tr> <tr><td> </td><td></td><td></td></tr> <tr><td> </td><td></td><td></td></tr> </table>	name	Values	\$company	file		\$file	names		\$filename.array						
name	Values	\$company														
file		\$file														
names		\$filename.array														
Error Message																
Message ID																
Add	true															
Update	true															
Delete																
Before	true															
Display																
Initial																

Creating the dynamic subform from the company record

1. To create a dynamic sub-format, log into ServiceCenter as a SysAdmin user, go to Database Manager and search for records in the company table.
2. In the company form perform a true search. The following screen is displayed:

Company Code	Company Name	Customer ID	Customer Since	City	State	Country
DEFAULT	Default Company	00000001	12/31/98 08:...			
HP	Hewlett-Packard	00000002	12/31/84 10:...	San Diego	CA	USA
ACME	ACME, INC.	00000003	12/31/98 08:...	San Diego	CA	USA
GFNFRTCOM	Genericom Computer Resources	00000004	05/31/95 10:...	Chicann	TI	USA

Company Information

Customer ID: 00000002

Company Code: HP Contacts

Company Name: Hewlett-Packard

Default SLA for Company: HP Platinum

Customer Since: 12/31/84 10:00:00

Address: 16399 W. Bernardo Drive

City/County/Post Code: San Diego CA 92127

Country: USA

Phone Number: (800) 638-5231

Fax Number: (858) 481-1751

Service Manager: MAX.MANAGER

Service Delivery Manager: MANAGER1

Show Company in Multi-Company Lists: Yes

Always Show Company in Lists: Yes

MANDATORY Affected Item

Labels	Fields	Types	FileNames	ForecolorC...	For...	VisibleCond	Visible	ReadonlyCond	Readonly
	Field1	specific.field2	probsummary		7		1		0

3. Click the record for the company for which you want to create or modify a sub-format. Scroll down to the Custom Fields table.
4. Make any changes in the Custom Fields table. You can add fields and labels, and change their types to Text (Default), Fill, Multitext, and Checkbox.
5. Click either **Save** or **OK** to save the record.
6. Go to the Forms Designer.
7. Search for the form called company.<company_name>.<tablename>.subformat.

Forms Designer: company.hp.probsummary.subformat

Field1

You can perform the preceding steps on all other company records, following these rules:

- Text can be put anywhere in fields that do not have the Array datatype.
- Multitext should be placed at the end, and input can be any Character Array field.
- Fill can be used for fields that are associated with a field in a link record.
- Checkboxes can be used for any Boolean field.

If the table of custom fields remains empty, a sub-format containing only the title is created, this helps avoid error messages when tickets for that company are displayed.

If a dynamically created form is changed in Forms Designer, these changes are undone whenever the company record is updated.

Appendix A

Checklist of records to modify to allow for this functionality

Service and Incident Management

File	Record	Notes
scmandant	As needed	
operator	As needed	
scsecuritygroup	As needed	
wizard	MSP Switch Company	Added to allow for company selection with calling the Process.
wizard	MSP Switch Company 2	Added to allow for company selection without calling the Process.
wizard	MSP Reset Company	Added to allow for reset of company selection.
Process	company.specific.incident	Sets mandant values and the sub-format name, and calls the open application.
Process	company.specific.incident2	Sets mandant values and the sub-format name, does not call the open application.
Process	cc.qopen.create	Sets mandant values and the sub-format name.
Process	cc.create	Sets mandant values and the sub-format name.
Process	us.screlate	Resets mandant values.
Process	im.view.init	Sets the sub-format name.
Process	im.new.incident	Calls the MSP Switch Company 2 wizard.
format	company.prompt.g	Selects the company to apply to this incident.
format	company . <company> . <tablename> . subformat	For company- and table-specific sub-formats
format	apm.quick	Adds a tab for a company-specific sub-format.
format	IM.template.*	Adds a tab for a company-specific sub-format.
dbdict	probsummary	Adds fields for company-specific fields.
dbdict	incidents	see above (if needed)
dbdict	company	Add custom.fields arrayed structre
formatctrl	login.DEFAULT	Adds the calculation: \$G.orig.secgroup=security.group in \$file1.
displayscreen	scm.advanced	Adds the initialization expression: \$wizard.name=" MSP Switch Company"
Object	probsummary	Calls the wizard \$wizard.name instead of the incident open application.
link	incidents	Allows the display of categories for the contact's company only. In ServiceCenter version 6 add the following Post Link Expressions to the contact.name line: \$L.security.token=str(toupper(company in \$File)); \$L.void=mandant(2,

		{L.security.token})
menu	IM	Calls wizard.run instead of apm.first.

Additional steps to perform on Change Management

Table	Record	Notes
Process	cm.update	Sets the sub-format name.
Process	cm.open	Calls a Switch Company wizard.
dbdict	cm3r	Adds fields for company-specific fields.
Object	cm3r	Calls the wizard \$wizard.name instead of the change open application.

For more information

Please visit the HP Software support Web site at:

www.hp.com/go/hpssoftwaresupport

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

HP Software online software 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 valued 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:

www.hp.com/go/hpssoftwaresupport/new_access_levels

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

www.hp.com/go/hpssoftwaresupport/passport-registration

© 2007 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.

HP and Service Manager are registered trademarks of Hewlett-Packard Development Company, L.P. JavaScript is a registered trademark of Sun Microsystems, Inc. in the United States and other countries.

01/2008

