Teams in Business Process Manager V8.5, Part 3: Managing teams

Gabriel Dermler (gabriel.dermler@de.ibm.com)
Senior Software Engineer
IBM

Markus Reichart (markus.reichart@de.ibm.com)
Software Architect
IBM

Thomas Bernhardt (thomas.bernhardt@de.ibm.com)
Development Team Lead
IBM

Thomas Duerr (thomas.duerr@de.ibm.com)
Software Engineer
IBM

19 February 2014

Part 3 of this series describes how to manage teams at runtime. It shows how to use the administrative tools provided as part of IBM® Business Process Manager V8.5 and it demonstrates how to implement services that perform automated team administration. For the modeling aspects of teams or how to build dynamic teams, see the previous articles in this series.

Introduction

A team definition is created as part of a process application or toolkit. Team definitions are deployed to a runtime environment when a snapshot of a process application that contains the team definition is deployed.

After deployment, you can use the Process Admin Console to add users or groups to or remove them from teams to react to changing business conditions. You can also change the assigned Managers team. However, if you remove the Managers team, the team and its tasks can no longer be managed using the Team Performance dashboard in Process Portal.
You can extend the administrative capabilities offered by the Process Admin Console by implementing custom services for team administration.

In Part 1, you have learned about the various modeling aspects of teams. In Part 2, you have built services to dynamically construct teams at runtime. This article, Part 3, describes how to use the Process Admin Console to manage teams and how to develop custom administration services.

Administering teams using the Process Admin Console

Members of the BPM security group can use the Process Admin Console to manage team definitions in snapshots of installed process applications. To see the teams in a snapshot, log in to the Process Admin Console, go to the Installed Apps tab, select a snapshot, then click the Team Bindings tab as shown in Figure 1.

Figure 1. Inspecting the Team Bindings

The Team Bindings page shows the current runtime definition for each team and whether the team is automatically refreshed. The definition includes a:

- Set of users
- Set of groups
- Team of managers

You can change the team's runtime definition by adding or removing users and groups, or by assigning a team of managers. Note, that for dynamic teams, an additional flag Automatic Refresh is displayed.

Part 2 in this series of articles on the IBM BPM V8.5 team functionality described how to define and implement dynamic teams using team retrieval services. Dynamic teams are re-evaluated at runtime, for example, to assign a task. To prevent administrative changes from being overwritten, the Process Admin Console contains an Automatic refresh enabled check box for dynamic teams. When the runtime definition of a dynamic team is manually changed, the check box is
automatically cleared and evaluation of the team by the team retrieval service is suspended. To switch back to the automatic refresh mode and to re-enable team evaluation, select the Automatic refresh enabled check box as shown in Figure 2.

**Figure 2. Enabling automatic refresh**

![Image showing process admin console](image)

When you re-enable the refresh mode, the team definition is refreshed with the latest information.

**Developing custom services for administering teams**

Consider developing custom services if you want to extend the team administration capabilities provided by the Process Admin Console, for example, to provide the following functionality:

- Custom user interfaces (UIs) for team administration
- Sophisticated team look-up mechanisms
- Automated services for periodically refreshing teams

**Example 1: Administrative service for refreshing teams**

As an alternative to the functionality provided by the Process Admin Console for team administration, you can define custom user interfaces that extend the Process Admin Console.

To provide a custom user interface, define a human service in Process Designer and expose it as an administration service to a team. Only members of the specified team can see and use the custom administration service in the Process Admin Console.

**Figure 3. Exposing a team retrieval service as administrative service**

![Image showing process admin console](image)
From a modeling perspective, an administration service is the same as any other coach-based human service. You model the user interface and the logic it needs to present the required data to the user, and to react to input from the user.

Figure 4 shows how to model an administration service for refreshing a team. The current runtime team definition, including any administrative changes, is overwritten with the original modeled team definition, if Automatic Refresh is enabled. For dynamic teams, the service that evaluates the team is triggered, and the team is overwritten with the result of the service call, if Automatic Refresh is enabled.

**Figure 4. Modeling a team retrieval service**

The server script is called with the team name that was entered, and it refreshes the team definition as shown in Listing 1.

**Listing 1. Implementing the Team Refresh Script**

```javascript
// get the team
var team = tw.system.org.findTeamByName(tw.local.teamName);

// refresh the team
if (team!=null) {
    team.refresh();
}
```

Figure 5 shows the coach that accepts the name of a team.

**Figure 5. Modeling a coach**

The server script is called with the team name that was entered, and it refreshes the team definition as shown in Listing 1.
Figure 6 shows how the service is made available in the Process Admin Console. To see and to invoke the service, users must be members of the team that was specified when the service was exposed.

**Figure 6. Calling an exposed administrative service**

![Image of Process Admin Console with Team Refresh Administration option]

**Example 2: Agent-driven service for periodically refreshing teams**

Members and managers of dynamic teams are automatically re-evaluated by the team retrieval services when the teams are used at runtime for task assignment.

If nothing happens in the system to trigger a team refresh, then the team information can become stale. The information can become stale, for example, if the team retrieval service for the team uses dynamic parameters to determine the team members and managers. Examples of dynamic parameters are the time of the day, the overall amount of work assigned to a team, or the current stock price of the company. In such cases, periodically refreshing teams is useful to ensure that the right people get the right tasks on their work list.

To refresh the team, a service is needed that does the work and a mechanism that invokes the service in a recurring fashion. To accomplish these requirements, you can use an undercover agent (UCA) to periodically trigger the service.

Figure 7 shows an UCA called Periodic Team Refresh UCA. The UCA triggers the Team Refresh Service every day at midnight.
The invoked service is an integration service that includes a JavaScript element shown in Listing 2.

### Listing 2. Refreshing all teams

```javascript
// get all teams
var allTeams = tw.system.org.getAllTeams();
// refresh teams one by one
for (var i=0; i<allTeams.length; i++) {
    var team = allTeams[i];
    team.refresh();
}
```

A more realistic implementation would consider additional aspects, for example, limit the number of teams refreshed at one time to a number that avoids overloading the system.

You can use the Process Admin Console to enable or disable the agent-triggered service. If disabled, the agent service does not trigger the associated Team Refresh Service as shown in Figure 8.
Figure 8. Checking the scheduling of the UCA

Conclusion

This article discussed the runtime management aspects of teams. It showed how to use the Process Admin Console to manage teams and how these capabilities can be easily extended by defining custom services. This article concludes the series on IBM BPM V8.5 team functionality.
Resources

- IBM Business Process Manager Information Center: Creating a team
- IBM Business Process Manager 8.5 Information Center: Configuring runtime teams
- IBM Business Process Manager Information Center: Adding functions to the Process Admin Console using services
- IBM Business Process Manager Information Center: Understanding and using undercover agents
- developerWorks Business Process Management zone
About the authors

Gabriel Dermler

**Gabriel Dermler** is a Senior Software Engineer on the Business Process Manager team in Böblingen, Germany. He has been involved with the design and implementation of Process Management and Portal solutions for many years, with one focus area in user and security management. Prior to that, Gabriel spent three years with IBM Research working on Quality of Service issues for IP telephony and web offerings. He holds a PhD degree in Computer Science from the University of Stuttgart, Germany.

Markus Reichart

**Markus Reichart** is a member of the IBM WebSphere Business Process Management architecture team at the IBM Research and Development Lab in Böblingen, Germany. He has twelve years of experience as a developer and architect in the areas of Business Process Management for WebSphere MQ Workflow, WebSphere Business Integration Server Foundation, WebSphere Process Server, and IBM Business Process Manager. His architectural responsibilities include administration and operations and for IBM Business Process Manager 8.5, Work Management. Markus holds a Master's degree in Computer Science from the University of Stuttgart and joined IBM Germany in 2001.

Thomas Bernhardt

**Thomas Bernhardt** is Team Lead of the development team for IBM Business Process Manager, focusing on the implementation of team concepts. He has worked for 17 years on IBM BPM and its predecessor products, both for human-centric and workflow-centric features.

Thomas Duerr

**Thomas Duerr** is a Software Engineer at the IBM Research and Development Lab in Böblingen, Germany. He has seven years of experience as a developer in the areas of WebSphere Process Server and IBM Business Process Manager. Currently, he is part of the development team for the IBM BPM offering and focuses on integration
with Enterprise Content Management Systems. Thomas holds an Engineering degree in Information Technology and joined IBM in 2006.