Using Parallel Dynamic Block and Dynamic User Assignment
© Copyright 2005 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.
<table>
<thead>
<tr>
<th>Type Style</th>
<th>Represents</th>
<th>Icons</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example Text</td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.</td>
<td><img src="image" alt="Caution Icon" /></td>
<td>Caution</td>
</tr>
<tr>
<td>Example text</td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
<td><img src="image" alt="Example Icon" /></td>
<td>Example</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
<td><img src="image" alt="Note Icon" /></td>
<td>Note</td>
</tr>
<tr>
<td>Example text</td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
<td><img src="image" alt="Recommendation Icon" /></td>
<td>Recommendation</td>
</tr>
<tr>
<td>Example text</td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
<td><img src="image" alt="Syntax Icon" /></td>
<td>Syntax</td>
</tr>
<tr>
<td>&lt;Example text&gt;</td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Business Scenario

You want to create a poll amongst specified users of the portal. You do this by creating and starting a process.

Introduction

In Guided Procedures there is a special block type, called Parallel Dynamic Block (PDB). As with standard parallel blocks, this block type can be processed at runtime by several users simultaneously; however, it has a different structure. The PDB always contains one subordinate block. The system creates as many instances of the subordinate block as there are rows in the input table of the PDB.

Process designers use the PDB if they do not know at design time how many instances will process the action of the PDB’s subordinate block in parallel. The name of this block type is derived from the fact that the system determines the number of block instances dynamically at runtime. With standard parallel blocks, the system behaves differently in that it uses a fixed number of block instances at GP design time, depending on the number of actions and blocks that make up the block.

These properties make the Parallel Dynamic Block an excellent candidate for building polls. In this tutorial you create and execute a PDB.

General Prerequisites

You need to have CAF GP installed.

Applicable Releases

This tutorial is compatible with the following release: Beginning with SAP NetWeaver '04s SPS6".

Disclaimer

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.
The Step-By-Step Solution

Create the Poll

1. Start the **Guided Procedures Design Time**. Navigate to the directory in which you want your process to be created and click **Create Process**.

2. Enter a name for the process.

3. Create a sequential block by clicking **Create New**.

4. Select the newly created block and name it “InitPollBlock”.

5. Choose the symbol for the block.
6. Click the newly created block and create an action in it by choosing Create New again.

7. Select the created action and enter a name for it. Click on it again to accept the changed name.

8. Click on the newly created action, name it, and create a callable object in it by choosing Create New again.
9. Select **Data Input Form** from the **Data Forms** group as the type. Enter the name and description. Click **Next**.

10. Define a structure list as output parameter. To do so, click **Insert New** and enter the name. Define the technical name “InputStructure” in the displayed row. Define **Structure** as **object type** and mark the checkbox in the **List** column.

11. Add two children with technical names **user** and **question** by clicking on **Insert Child**. Click **Next** to continue.
12. Click **Finish** to complete creation of the callable object.

13. Select the row containing the process. Click on **Create New** to create a parallel dynamic block.

14. In the Block Type Selection panel select type **Parallel Dynamic Block**.
15. Select the newly created parallel dynamic block and give it a name (“PollBlock”).

16. Choose ☑️ to accept the new block.

17. Create a sequential block in it by clicking on Create New again.

18. Select the new created block, name it “SubBlock”, and choose ☑️.
19. Select the sequential block you just created and click on **Create New** again to create an action in it.

20. Select Action2, name it “PolAction”, and choose [button].

21. Select the created action and click on **Create New** again to create a callable object in it. Select **Input** as the type of the callable object. Enter a name and description. Click on **Next** to continue.

22. Add two input parameters with technical names **user** and **question** by clicking on **Insert New**. Click **Next** to continue.
23. Add an output parameter named **answer** by clicking on **Insert New**. Click **Next** to continue.

24. Click **Finish** to complete creation of the callable object.

25. Click on the row containing the process. Click on **Create New** to create a sequential block.
26. Click on the newly created block, give it a name, and create an action in it by choosing Create New again.

27. Click on the newly created action, give it a name, and create a callable object in it by choosing Create New again.

28. Select Data Display Form from the Data Forms group as type. Enter a name and description. Click Next.
29. Add a structure list as input parameter by clicking on **Insert New**. Enter “OutputStructure” as technical name. Enter the name and define the **Object type** as **Structure**. Mark the checkbox in the **List** column. Add an output parameter named **answer** by clicking on **Insert New**. Click **Next** to continue.

Note: Although it is an input structure, the parameter is named **OutputStructure**, indicating that its content is derived from the first PollBlock.

30. Click **Finish** to complete creation of the callable object.

31. The flow of the Guided Procedures process is now defined. It should look like the process shown on the right.
32. Set up the dynamic role assignment. Select the sequential block in the parallel dynamic block and navigate to the Roles tab on the bottom pane.

33. Mark the checkbox **Filled From Context Parameter** and select context parameter **user** from the dropdown list.

34. Click on the row containing the **Parallel Dynamic Block** and check its input and output parameters by clicking the Parameters tab. The output should look like the display on the right.
35. You must now consolidate the context variables for this process. Click on the row containing the Process and click the Parameters tab on the bottom pane.

36. In the Context Parameter list, mark the checkboxes for “InputStructure” and “MultilineInputStructure”. Enter “PollInput” in field Consolidate To, and click Go. The selected parameters are consolidated under the given name.

- PollInput
  - InputStructure
    - question
    - user
  - Multiline Input Structure
    - question
    - user

**Note:** In this case consolidation only works properly if the structures to be consolidated have the same structure and the technical names of the elements of one structure correspond to the technical names of the elements of the other structure.

37. Consolidate “OutputStructure” and “Multiline Output Structure” under the name “PollOutput”. After consolidation, the parameters should be as shown on the right.
38. Now consolidate the roles. To do so, click on the Roles tab. Select both roles in the list by marking them. Enter the name “Processor” in the Consolidate To input field and click Go.

39. Define the Role Type of the newly created role as Initiator.

40. Click on the Built-in Roles tab and define the role type of the built-in roles as Initiator. The process is now complete. To execute it, activate it by clicking on Activate (↑).
Executing the Poll

41. You can start the process directly from design time by clicking on the Instantiation tab, which is located at the far right in the bottom pane. Choose Generate Instantiate URL and click on Open Instantiate Application to open the instantiation window.

Note: There are also other means of instantiation.

42. Upon successful instantiation a window appears in which you can change the initiator and process name. Choose Initiate to start the process.

43. The first step of the created process appears. Click on Add Row to address a user with a question.
44. In this example, two rows were added. User names must be provided in a special format, unique ID. When you have finished preparing the poll, click on **Submit** to launch it. The poll appears in the GP worklist of each addressed user.

**Note:** You can find this ID in **User Administration -> Identity Management** by searching for the given user. The ID is displayed under the **General Information** tab.

45. The addressed user can see that new GP actions are waiting for intervention by choosing **Guided Procedures -> Run Time** after logon.

46. Click on **Tasks** that require my action to display the list of pending tasks. Select the row containing the action that is part of the process.
47. You can continue the poll by responding to the action.

48. When all participants of the poll have responded, the process returns to the initiator and appears in the initiator’s GP worklist. The initiator can open it, display all answers and click OK if the process is complete.