



# Using Room Extensions

## SAP NetWeaver Product Management

## Introduction to Room Extensions

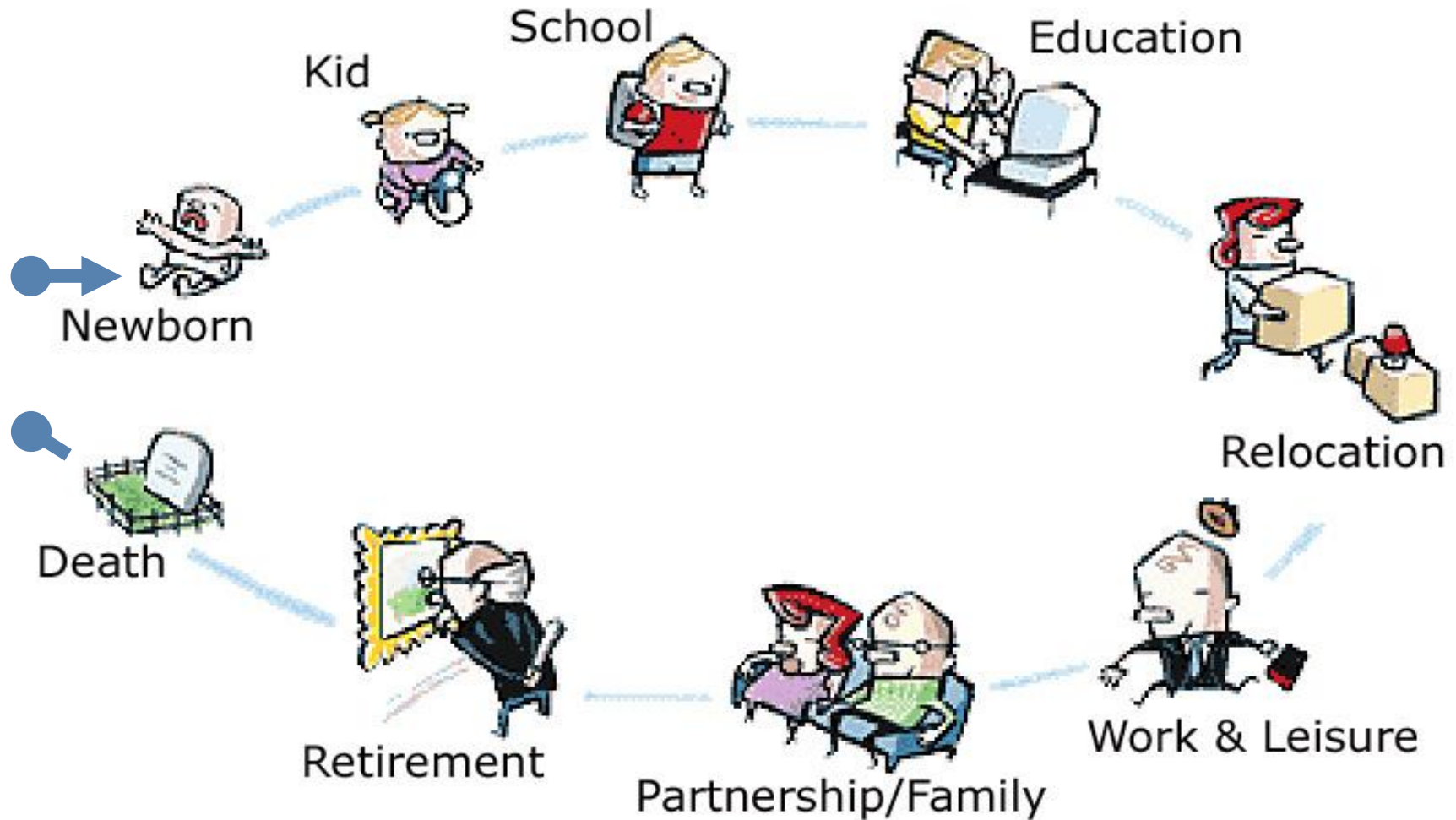
Mapping Plan

Example

How to develop an Extension

Value Set Provider

# Human Life Cycle



## Purpose of Room Extensions

- **During their life cycle, rooms and room parts pass through defined points:**
  - **Creation of a room or room part**
  - **Deletion of a room or room part**
  - **Adding of a room part to the room**
- **These exact points of time can be used to plug in specific code**
  - **This code can be written according to company-specific needs**
  - **This specific code is called an extension and the points in the life cycle are called extension points.**
- **Extension purposes:**
  - **Extensions define for instance the repositories for iViews in rooms and room parts – they are mandatory integral parts of SAP's template definitions**
  - **Extensions can be used for connecting to backend systems, to set permissions or to provide or concatenate information.**

## Overall objective of Room Extensions

### (Functional) Extensibility

- Integration of non standard systems at customer site
- Functional enhancements

## Advantages of Room Extensions

### Flexibility

- Customers can add any Java coding without changing and touching SAP code

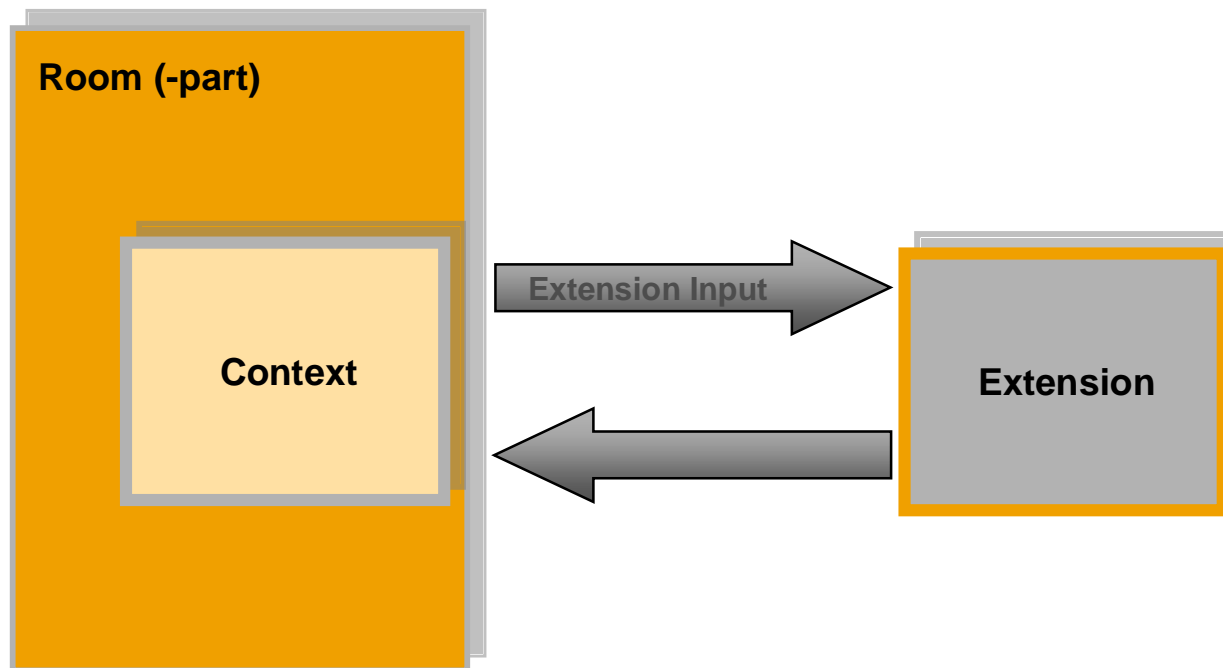
### Stability

- Customer code should not be changed or overwritten when updating a system (e.g. via patches or hot fixes)

# What is an Extension?

## Extension

An extension is a *portal service* that adds functionality to a SAP Collaboration Room. It defines input and output parameters, which are mapped at *design time* and processed at *run time*, when dedicated *extension points* are reached. Extensions are similar to plug-ins or customer exits.



Data flow of an extension, at a dedicated extension point

# Where do the Parameters come from?

## Context

Set of parameter values that belong to a room or a room part.  
The parameters may be:

- **Constants** defined by the template creator at *design time*
- **External parameters** prompted from the user at *run time*
- **System Parameters** added by the room at *run time*
- **Extension output** processed at *run time*

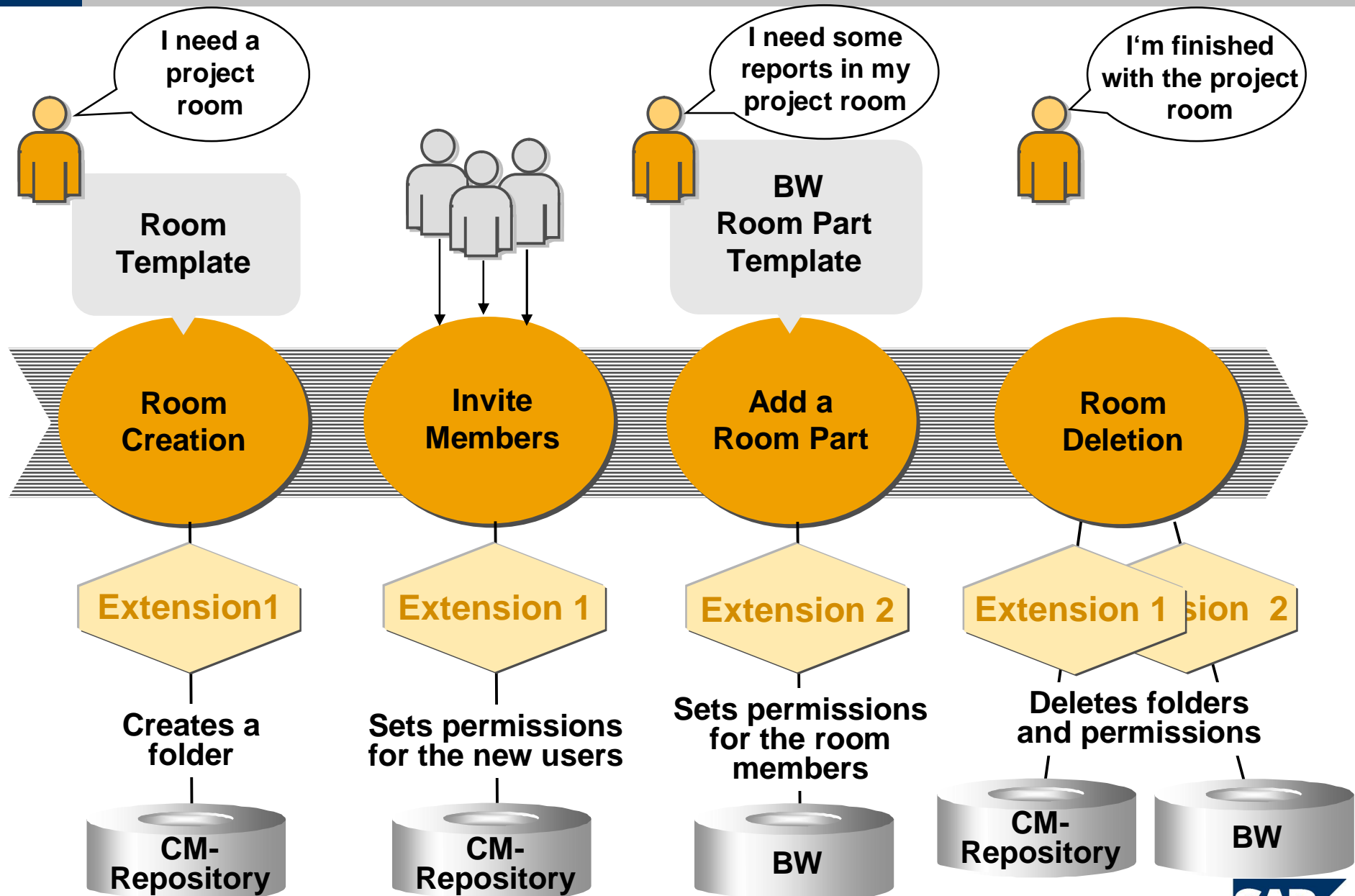
## Design Time

The creation of a room template or a room part template

## Run Time

The life cycle of a room that begins with the creation and ends with the deletion

# Collaboration Room Life Cycle (Example)





## Extensions provided by SAP

Id	Category	Description
cmRoomExtensionSP2FP	room	Creates, handles, and deletes CM workspaces for rooms and room parts. Also organizes the entry points to allow copy and move activities between the workspaces of a room.
roomPropertyReader	room	Exposes properties from the system parameter ROOM_INFO.
roompartPropertyReader	room	Exposes properties from the system parameter ROOM-PART_INFO.
concat	common	Concatenates two strings.
configReader	common	Exposes properties of configurables in the Configuration Framework.

# How are the Parameters assigned?

## Parameter Mapping

Step at *design time*, where the template creator assigns extension parameters to parameters of the room or the room part *context*.

The screenshot shows the 'Edit Extension' dialog box with the 'Parameters' tab selected. It contains two tables: 'Input Parameters' and 'Output Parameters'.

**Input Parameters**

Name	Parameter Mapping	Value Help	Fixed Value
Store *	Use Value Help	Documents	
Template	Use Value Help	No Value	
Workspace Name	Fixed Value	No Value	Documents
Access	Use Value Help	private	

**Output Parameters**

Name	Mapped	Room Part Parameter
Workspace	<input type="checkbox"/>	
Workspace Path	<input checked="" type="checkbox"/>	private_docs_path

Buttons: Save, Cancel

Introduction to Room Extensions

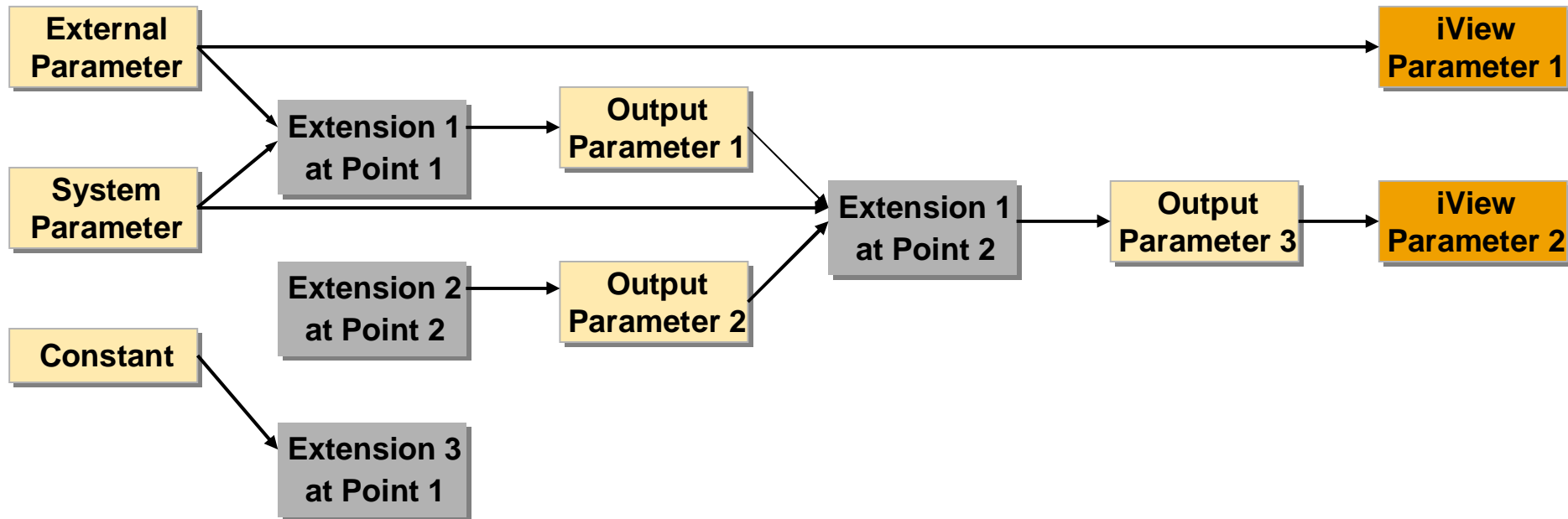
**Mapping Plan**

Example

How to develop an Extension

Value Set Provider

# Chaining Extensions - Mapping Plan



## Parameter-Mapping Plan

External parameters, system parameters, output parameters, and constants are used as input parameters of extensions at a dedicated extension point.

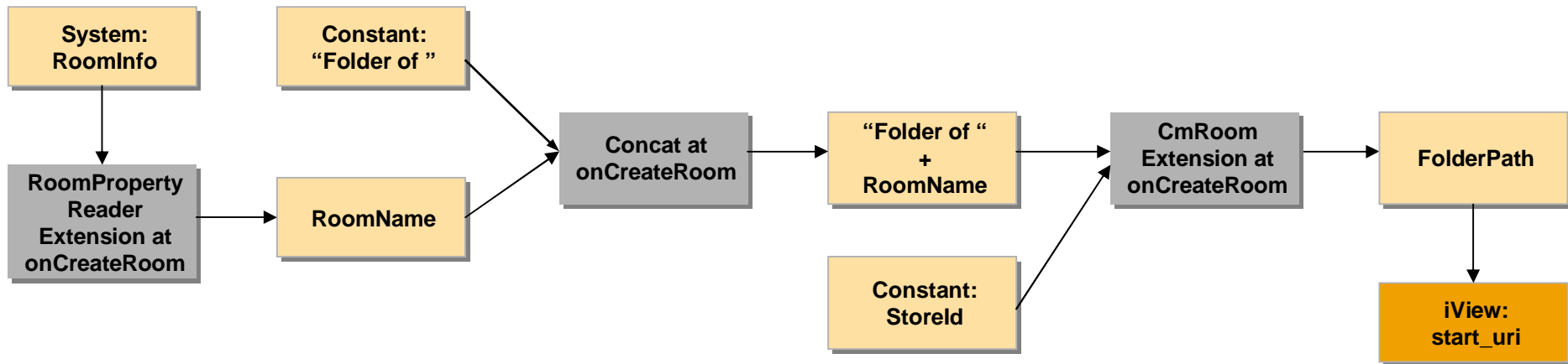
# Chaining Extensions - Mapping Plan Example

## Example

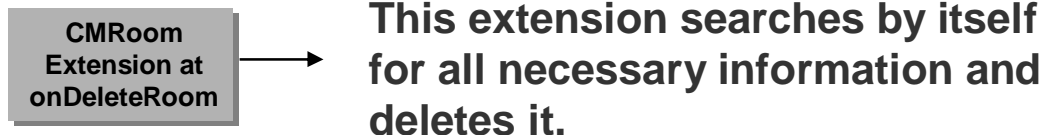
The template creator wants the room to create a document folder that should be named “Documents of %s”. At “%s” the creator wants to have the name of the room the folder belongs to.

The creator uses the `RoomPropertyExtension` to extract the room name from the system parameter `RoomInfo`. The `Concat` extension allows concatenating the room name and the string constant “Folder of “. The result is given to the `CmRoomExtension`, which creates the folder and outputs the path that is mapped to the `start_uri` of an `iView`.

## Creation Process

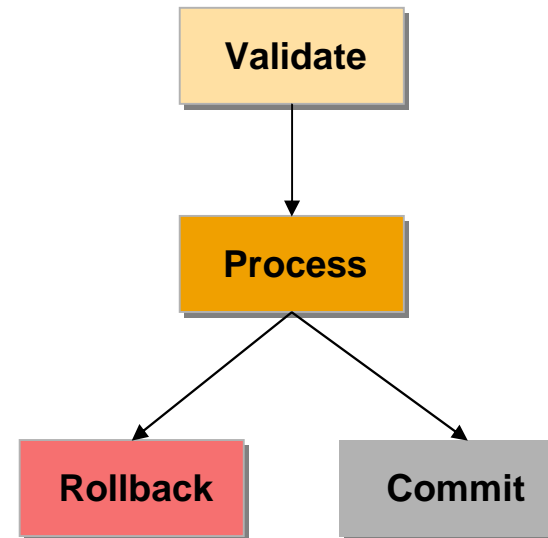


## Deletion Process



## Run Time Execution

When an extension point is processed, first the validate method of all extensions is called. If one extension fails, the processing of the extension point terminates on error. If all extensions can validate, the process method is called in a second step. If one extension fails to process, all extensions are rolled back, otherwise all commit.



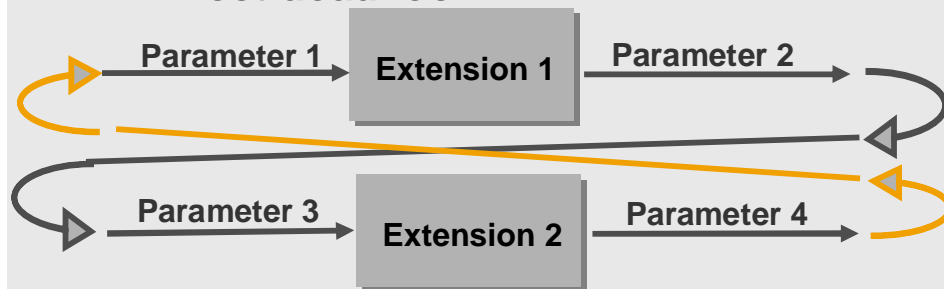
## Typical Error Message

cmDiscussionExtension used by Testroom cannot process extension point ON\_SET\_OWNER\_OF\_ROOM because parameter folderPath is missing

# Extension – Dead Locks!!!

## Common errors, which must be avoided

### 1. Direct dead lock



The output parameter 4 supposed to be also the input parameter 1 which is not possible.

**Error message: parameter missing**

### 2. Time dead lock



The input parameter 3 is supposed to be available before it has been created.

**Error message: parameter missing**

Introduction to Room Extensions

Mapping Plan

**Example**

How to develop an Extension

Value Set Provider



## Cm Room Extension

**Creates, handles and deletes CM folders for the iViews of rooms and room parts. Also organizes the entry points to allow copy and move activities between the folders.**

# CM Room Extension: Extension Points

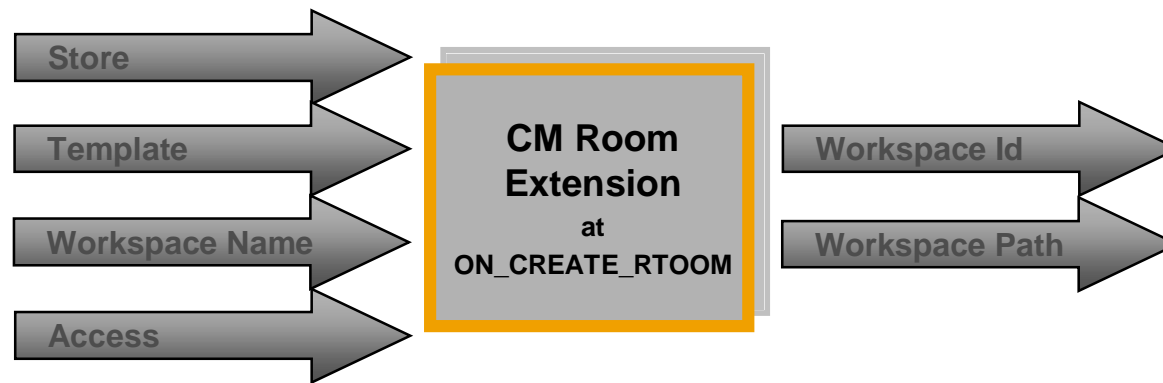
Name	Extension	Extension Point
Documents	cmRoomExtensionSP2FP	ON_ADD_ROOMPART_TO_ROOM ON_CREATE_ROOMPART ON_DELETE_ROOMPART ON_REMOVE_ROOMPART_FROM_ROOM

Add Extension

Save Cancel

- If the extension is used in the room, it must be configured for the extension points **ON\_CREATE\_ROOM** and **ON\_DELETE\_ROOM**.
- If the extension is used in the room part, it must be configured for the extension points **ON\_CREATE\_ROOMPART**, **ON\_DELETE\_ROOMPART**, **ON\_ADD\_ROOMPART\_TO\_ROOM** and **ON\_REMOVE\_ROOMPART\_FROM\_ROOM**.

## CM Room Extension: Parameters



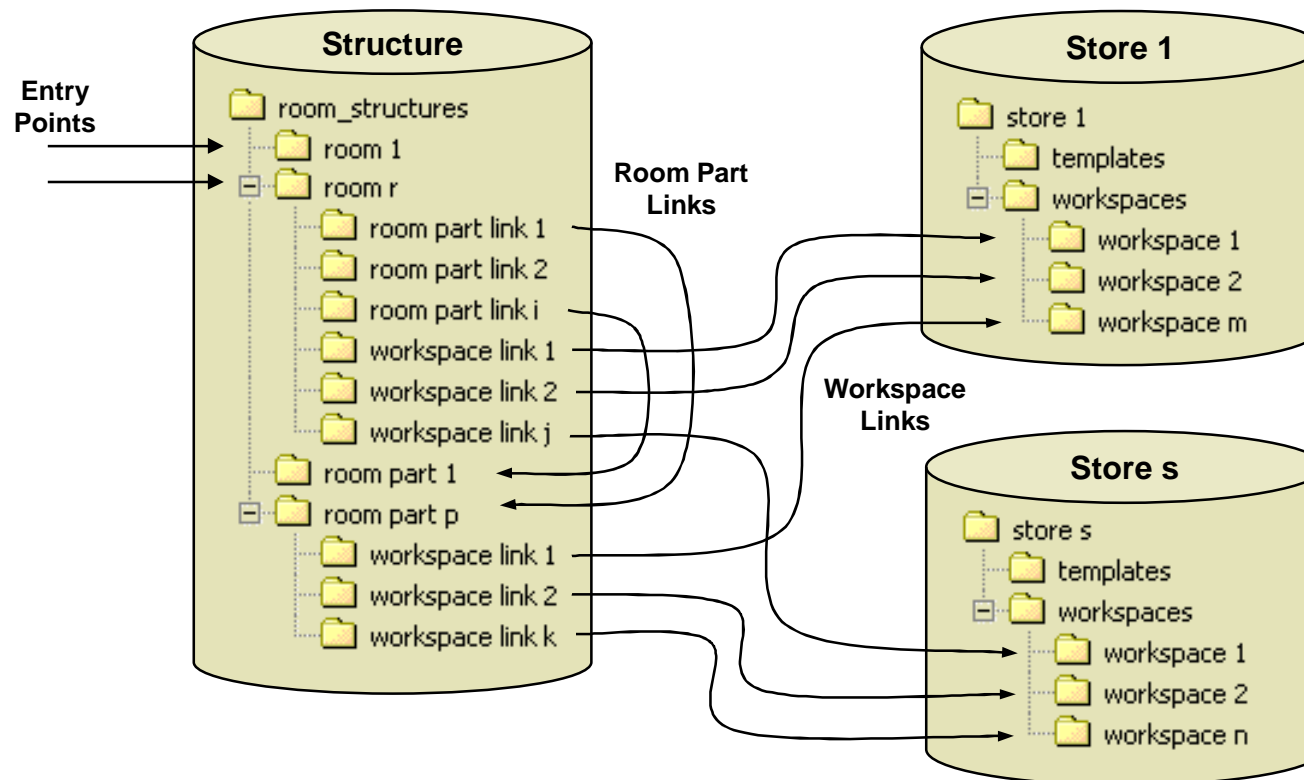
- **Store:** Preconfigured folder in the Repository Framework, where the extension will create the new folder that it will handle. The configuration is located in “System Administration -> System Configuration -> Knowledge Management -> Collaboration -> Room -> Extensions -> Cm”.
- **Template:** Preconfigured folder that is copied to a store to provide initial content.
- **Workspace Name:** Name of the folder that the CM Room Extension will create and manage.
- **Access:** determines, if the portal group “Everybody” is granted read access on the created folder.
- **Workspace Id:** Id of the created folder.
- **Workspace Path:** Path of the created folder.

## CM Room Extension: How it works

- **ON\_CREATE\_ROOM:** The CM Room Extension creates a folder in the configured store, copies the template content to that folder and grants full access for the main user group of the room and read access for the group “Everybody” if access “Public” is configured. It also adds the room main group to the owner groups of the folder. Finally it adds a folder link to the room in the room structure.
- **ON\_DELETE\_ROOM:** The CM Room Extension removes the folder link from the room structure and deletes the folder.
- **ON\_CREATE\_ROOMPART:** The CM Room Extension creates a folder in the configured store, copies the template content to that folder and grants read access to the group “Everybody” if access “Public” is configured.
- **ON\_ADD\_ROOMPART\_TO\_ROOM:** The CM Room Extension grants full access to the main user group of the room and adds this group to the owner groups of the folder. It adds a folder link to the room part in the room structure.
- **ON\_REMOVE\_ROOMPART\_FROM\_ROOM:** The CM Room Extension removes access for the main user group of the room and removes ownership. It also removes the folder link in the room structure.
- **ON\_DELETE\_ROOMPART:** The CM Room Extension deletes the folder.

# CM Room Extension: Output

In the Repository Framework, the CM Room Extension creates a linked structure of resources.



Introduction to Room Extensions

Mapping Plan

Example

**How to develop an Extension**

Value Set Provider

# Extension APIs – How to develop an extension

- 1. Create a new Java portal project**
  - Including a Java class for the extension
- 2. Configure**
  - portalapp.xml and
  - extension registry
- 3. Include extension API**
- 4. Provide a resource bundle to make it displayable**

## 5. Define

- extension points
- input- and output parameters

## 6. Implement extension methods

- validate
- process
- rollback
- commit



## ■ Extension APIs – Configure portalapp.xml

- **The extension is a portal service**
  - **A service entry in the portalapp.xml file is required**
  - **Service name can be chosen freely as long as it fulfils the Java naming conventions**
  - **Generic\_classloader\_registration must be activated**
  - **Define generic service key defined (must be unique)**

## Example

```
<service name="SampleExtensionService">
  <service-config>
    <property name="className" value="com.sample.SampleExtension"/>
    <property name="startup" value="true"/>
  </service-config>
  <service-profile>
    <property name="generic_classloader_registration" value="yes"/>
    <property name="generic_service_key" value="someSampleGuid"/>
  </service-profile>
</service>
```

See also commented code samples in [SAP Developer's Guide](#)

## Extension APIs – Configure extension registry

- **The extension must be registered in the extension registry**
  - Located in the configuration path Collaboration→Extension→Registry
  - A common choice is to take the name of the class for the id
  - The property service is a combination of
    - ◆ <par file name> concatenate <service name defined in the file portalapp.xml>
    - ◆ Please note that the par file name and the service name may differ from the package name and the class name
  
- **There are two possibilities**
  - Configure the extension registry manually or
  - Do deploy one with NW Development Studio (but only a workaround)

# Extension APIs – Configure extension registry

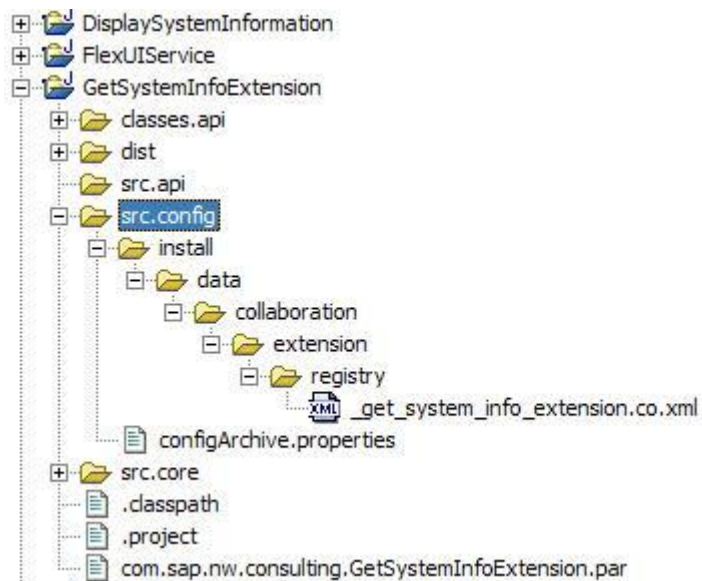
- Update the extension registry manually
  - System Administration → System Configuration → Collaboration → Extension → Registry → New!

The screenshot shows the SAP Extension Registry configuration interface. On the left, a navigation pane shows the path: Configuration → Collaboration → Extension → Registry. Below this, it lists 'Classes' including 'Extension (10)' and 'Value Set Provider (3)'. The main area is titled 'List of available extensions' and contains a table with columns for 'Id', 'Category', and 'Service'. Below the table are buttons for 'New', 'Duplicate', 'Edit', 'View', and 'Delete'. A 'View' dialog for 'cmRoomExtensionSP2FP' is open, showing fields for 'Category' (room), 'Service' (com.sap.netweaver.coll.appl.room.CmRoomExtensionSP2FP), and 'Active' (checked). A 'Help' button is located at the bottom left of the interface.

Id	Category	Service
<input type="checkbox"/> ProjectCreationWizard	room	com.siemens.pct.pmportal.proj_admin.collab.livelink.skey.roomext.ProjectC
<input type="checkbox"/> cmDiscussionExtension	room	com.sap.netweaver.coll.appl.room.CmDiscussionExtension
<input type="checkbox"/> cmRoomExtensionSP2FP	room	com.sap.netweaver.coll.appl.room.CmRoomExtensionSP2FP
<input type="checkbox"/> concat	common	com.sap.netweaver.coll.appl.room.Concat
<input type="checkbox"/> configReader	common	com.sap.netweaver.coll.appl.room.ConfigReader
<input type="checkbox"/> roomPropertyReader	common	com.sap.netweaver.coll.appl.room.RoomPropertyReader
<input type="checkbox"/> roomStatusExtension	roomsystem	com.sap.netweaver.coll.appl.room.RoomStatusExtension
<input type="checkbox"/> roompartExtension	common	com.sap.netweaver.coll.appl.room.RoomPartExtension

# Extension APIs – Configure extension registry

- Update the extension registry with your configurable
  - Create a configurable in the NW Development Studio (DEMO)



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- file generated on 1723, 20040629 -->
<!-- properties, etc. need to be added and adjusted -->
- <Configurable configclass="Extension">
  <property name="id" value="GetSystemInfoExtension" />
  <property name="service"
    value="com.sap.nw.consulting.GetSystemInfoExtension.GetSystem
  <property name="category" value="room" />
  <property name="active" value="true" />
</Configurable>
```

## Extension APIs – Include Extension API

- **Add libraries to your**
  - `portalapps/com.sap.netweaver.kmc.util/lib/kmc.util.core_api.jar`
  - `portalapps/com.sap.netweaver.coll.shared/lib/coll.shared.extension_api.jar`
  - `portalapps/com.sap.netweaver.coll.shared/lib/coll.shared.roomobject_api.jar`
- **Add a service reference in the portalapp.xml file of your project**

```
<application-config>
```

```
...
```

```
<property name="ServicesReference"  
          value="...,com.sap.netweaver.coll.shared, ..."/>
```

```
...
```

```
</application-config>
```

### ■ Resource bundles in your java class

```
private static final String BUNDLE = "com.sample.bundles.SampleBundle";  
private static final String ID = "sampleExtension";
```

```
public String getName(Locale locale) {  
    return ResourceBundle.getBundle(BUNDLE).getString("lbl_" + ID, locale); }
```

```
public String getDescription(Locale locale) {  
    return ResourceBundle.getBundle(BUNDLE).getString("dsc_" + ID, locale);}
```

### ■ Resource bundle is located in the same project as the extension

### ■ Example

- lbl\_sampleExtension=Sample Extension  
dsc\_sampleExtension=Sample with explanations how to create an extension

## ■ Extension points

- Steps in the life cycle of a room, where the extension is executed
- It depends on the purpose of the extension, which extension points are useful
- Provided in the class `RoomExtensionPoint`
- The extension provides a list of points it can handle

## ■ Example

```
public IExtensionPoint[] getExtensionPoints() {  
    IExtensionPoint[] result = {  
        RoomExtensionPoint.ON_CREATE_ROOM,  
        RoomExtensionPoint.ON_DELETE_ROOM, //... };  
    return result;  
}
```



## Extension APIs – Define Input-Output Parameter

- **An extension also defines a list of input- and output parameters**
  - **Dependent on the extension point (e.g. ON\_CREATE\_ROOM)**
  - **Example, an extension that creates a folder may require a parent path and output a folder id**
  
- **Input parameters**
  - **System parameters are contributed by the room (see next slide)**
    - ◆ **Example NEW\_OWNER\_OF\_ROOM is available only when ON\_SET\_OWNER\_OF\_ROOM is processed**
  - **Template parameters are defined in the template.**
    - ◆ **Only string parameters are allowed**

# Extension APIs – Define Input-Output Parameter

## ■ Extension points and system parameters added to the context (not complete)

	<i>Extension Point</i>	<i>Predecessor</i>	<i>System Parameter Ids</i>
A	ON_CREATE_ROOM		ROOM_INFO
B	ON_ADD_ROOMROLE_TO_ROOM	A	ROOM_INFO ADDED_ROOMROLE_TO_ROOM
C	ON_ADD_USERS_TO_ROOMROLES	A, B	ROOM_INFO ADDED_USERS_TO_ROOMROLES
D	ON_SET_OWNER_OF_ROOM	A	ROOM_INFO OLD_OWNER_OF_ROOM NEW_OWNER_OF_ROOM
E	ON_AFTER_CREATE_ROOM	A, B, C, D	ROOM_INFO
F	ON_CREATE_ROOMPART		ROOMPART_ID ROOMPART_TEMPLATE_ID ROOMPART_INFO
G	ON_ADD_ROOMPART_TO_ROOM	A, B, C, D, E, F	ROOM_INFO ROOMPART_INFO
H	ON_REMOVE_ROOMPART_FROM_ROOM	A, B, C, D, E, F, G	ROOM_INFO ROOMPART_INFO
I	ON_DELETE_ROOMPART	F	ROOMPART_ID ROOMPART_TEMPLATE_ID ROOMPART_INFO
J	ON_BEFORE_DELETE_ROOM	A, B, C, D, E	ROOM_INFO
K	ON_REMOVE_USERS_FROM_ROOMROLES	A, B, C	ROOM_INFO REMOVED_USERS_FROM_ROOMROLES
L	ON_REMOVE_ROOMROLE_FROM_ROOM	A, B	ROOM_INFO REMOVED_ROOMROLE_FROM_ROOM
M	ON_DELETE_ROOM	A, B, C, D, E, J, K, L	ROOM_INFO
N	ON_LOCK_ROOM	A, B, C, D, E	ROOM_INFO
O	ON_UNLOCK_ROOM	A, B, C, D, E, N	ROOM_INFO

# Extension APIs – Define Input-Output Parameter

## ■ Example

```
public IParameterInfo[] getInputParameterInfos(IExtensionPoint extensionPoint) {
    //ON_CREATE_ROOM
    if (RoomExtensionPoint.ON_CREATE_ROOM.equals(extensionPoint)) {
        IDisplayable parameter1Displayable = extensionFactory.createDisplayable(
            PARAMETER1_ID,
            RESOURCE_BUNDLE,
            "lbl_" + PARAMETER1_ID,
            "dsc_" + PARAMETER1_ID);

        IParameterInfo[] result = {
            extensionFactory.createInputParameterInfo(
                parameter1Displayable,
                PARAMETER1.CLASS,
                VALUE_SET_PROVIDER_ID,
                optional),

            extensionFactory.createSystemParameterInfo(
                PARAMETER2_ID,
                PARAMETER2.CLASS,
                mandatory),
            //... };

        return result; }
    //ON_DELETE_ROOM //...
    return null; }
```

- **To build up the IParameterInfo for a system parameter requires**
  - A parameter id
  - The class to validate the parameter value, e.g. it can be checked if a user input is an integer
  - A Boolean to define if the parameter is mandatory or optional
- **Template parameters are used in the template → require additional information**
  - A displayable to show the parameter in the template wizard
  - An optional value set provider id
    - ◆ to show the value set in a drop down list box

# Extension APIs – Define Input-Output Parameter

## ■ Extension output

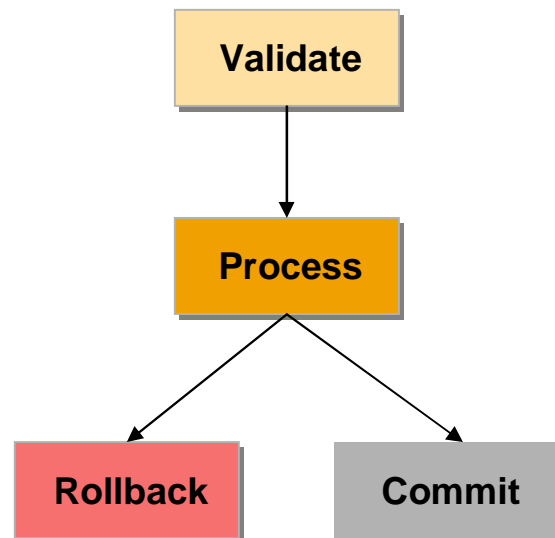
- Are parameters filled by extension results
- An extension can even use its own output as input for a later extension point

## ■ Example

```
public IParameterInfo[] getOutputParameterInfos(IExtensionPoint extensionPoint){  
    //ON_CREATE_ROOM  
    if (RoomExtensionPoint.ON_CREATE_ROOM.equals(extensionPoint)) {  
        IParameterInfo[] result = {  
            createOutputParameterInfo(PARAMETER3_ID,  
                                       PARAMETER3_CLASS),  
            //...};  
        return result;  
    }  
    //ON_DELETE_ROOM  
    //...  
    return null;  
}
```

## ■ Extensions and transactional behavior

- When an extension point is processed, first the validate method of all extensions is called.
- If one extension fails, the processing of the extension point terminates on error.
- If all extensions can validate, the process method is called in a second step.
- If one extension fails to process, all extensions are rolled back, otherwise all commit.



## ■ Validate Method

- Extension takes the input parameters from the context and throws an exception if they are not consistent.
- Nothing should be processed only validates the parameters.
- The validation also may enclose to check the availability of a used backend or a required configurable.
- Must not be implemented, but can cause a high performance effort, because avoidable errors must be rolled back over all extensions involved in an extension point.

## ■ Process Method

- Extension takes the input parameters from the context
- Processes them and
- Returns the output parameters as the result
- Everything processed in this method must be reversible, e.g. if the extension deletes something, it should copy it to a recycler or create a backup



### ■ Rollback Method

- Is provided with the same context as the process method.
- But it may be enriched with objects created during process.
- Task of the rollback is to undo everything of the process method, e.g. created objects are deleted, and deleted objects are restored, recycled or recreated.

### ■ Commit Method

- Finalizes the execution of the extension.
- It is not allowed to do any processing here.
- The extension must have fulfilled its purpose, even if commit would not be called.
- Commit just removes backups provided for the rollback that are no longer useful.

## Example for a process method

```
public IExtensionResult process(IExtensionPoint point, IExtensionContext context)
    throws ExtensionException {

    //ON_CREATE_ROOM
    if (RoomExtensionPoint.ON_CREATE_ROOM.equals(point)) {

        //get parameters
        String parameterValue1 =
            (String)context.getValue(PARAMETER1_ID);

        String parameterValue2 =
            (String)context.getOptionalValue(PARAMETER2_ID);

        //do the processing
        newObjectId = ...

        //...
    }
}
```

## Example for a process method

```
//enable rollback
context.putValue(PARAMETER3_ID, newObjectId);

//return result
IExtensionResult result =
    extensionFactory.createExtensionResult(IExtensionResult.OK);

result.putValue(PARAMETER3_ID, newObjectId); return result; }

//ON_DELETE_ROOM
//...
return null;
}
```

See also commented code samples in [SAP Developer's Guide](#)

Introduction to Room Extensions

Mapping Plan

Example

How to develop an Extension

**Value Set Provider**

## Definition

- Provides a set of allowed values for a dedicated parameter.
- This set can be used to show drop down list boxes in the UI or to validate a user input.

## Value set providers are - like extensions - portal services

- Value set providers are also displayed in the template wizards
- The first steps to develop a value set provider are identical to developing an extension
- IDs are provided, the same methods are implemented to make the value set provider displayable
- Value set providers defines a list of parameters, for which they exposes value sets.

## Example

```
public IParameterInfo[] getParameters() {  
  
    IParameterInfo[] result = {  
  
        createParameterInfo(PARAMETER1_ID,  
                            PARAMETER1_CLASS),  
        createParameterInfo(PARAMETER2_ID,  
                            PARAMETER2_CLASS),  
        //... };  
  
    return result;  
  
}
```

### **A Value Set Provider may require some parameters**

- **E.g. if the value set should be the days of a month, it would require the month and the year to calculate the output.**
- **Currently not supported.**

## Example

```
public IParameterInfo[] getRequiredParameters(IParameterInfo parameter) {
    IParameterInfo[] result = null;

    //PARAMETER1
    //no input required

    //PARAMETER2
    if (parameter2.getId().equals(parameter.getId())) {
        IParameterInfo[] result = {
            createParameterInfo(PARAMETER3_ID,
                                PARAMETER3_CLASS),
            //... };

        return result;
    }

    //PARAMETER3
    //...

    return null;
}
}
```



## Make the value set provider executable

### ■ Example

```
public IValueSet getValueSet(IParameterInfo parameter, IExtensionContext context) throws  
    ValueSetProviderException {
```

```
    //PARAMETER1
```

```
    if (parameter1.getId().equals(parameter.getId())) {
```

```
        //get input parameters
```

```
        String parameterValue2 = (String)context.getValue(PARAMETER2_ID);
```

```
        //compute the value set
```

```
        IDisplayable value1Displayable = extensionFactory.createDisplayable(  
            VALUE_ID1,  
            RESOURCE_BUNDLE,  
            "lbl_" + VALUE_ID1,  
            "dsc_" + VALUE_ID1);
```

```
        Object value1 = getValue(parameterValue2);
```

```
        ...
```

## Example (cont)

```
IParallelValue[] result = {
    extensionFactory.createParameterValue(value1Displayable, value1),
    extensionFactory.createParameterValue(value2Displayable, value2),
    //...
};
return extensionFactory.createValueSet(result, exclusive);
}
//PARAMETER2
...
return null;
}
```

See also commented code samples in [SAP Developer's Guide](#)

## Further Information

### → Public Web

- **NetWeaver Developer's Guide:**  
<http://www.sdn.sap.com/irj/sdn/devguide2004s> → Enabling User Collaboration
- **Commented Coding Samples:**  
[http://help.sap.com/saphelp\\_nw04s/helpdata/en/42/c30767feef1d74e10000000a1553f6/frameset.htm](http://help.sap.com/saphelp_nw04s/helpdata/en/42/c30767feef1d74e10000000a1553f6/frameset.htm)

### → SAP Service Marketplace

- **Role-specific Learning Maps (available for purchase):**  
<http://service.sap.com/rkt-netweaver> → Available OKPs for SAP NetWeaver

- 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.
- SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.
- These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

- Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.
- Die von SAP AG oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten.
- Microsoft, Windows, Outlook, und PowerPoint sind eingetragene Marken der 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, und Informix sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern.
- Oracle ist eine eingetragene Marke der Oracle Corporation.
- UNIX, X/Open, OSF/1, und Motif sind eingetragene Marken der Open Group.
- Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, und MultiWin sind Marken oder eingetragene Marken von Citrix Systems, Inc.
- HTML, XML, XHTML und W3C sind Marken oder eingetragene Marken des W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- Java ist eine eingetragene Marke von Sun Microsystems, Inc.
- JavaScript ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.
- MaxDB ist eine Marke von MySQL AB, Schweden.
- SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver und weitere im Text erwähnte SAP-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern weltweit. Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.
- In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden. Die vorliegenden Angaben werden von SAP AG und ihren Konzernunternehmen („SAP-Konzern“) bereitgestellt und dienen ausschließlich Informationszwecken. Der SAP-Konzern übernimmt keinerlei Haftung oder Garantie für Fehler oder Unvollständigkeiten in dieser Publikation. Der SAP-Konzern steht lediglich für Produkte und Dienstleistungen nach der Maßgabe ein, die in der Vereinbarung über die jeweiligen Produkte und Dienstleistungen ausdrücklich geregelt ist. Aus den in dieser Publikation enthaltenen Informationen ergibt sich keine weiterführende Haftung.