



S H A R E
Technology • Connections • Results

WebSphere Message Broker Patterns

David Coles – WebSphere Message Broker Level 3 Service,
IBM Hursley – dcoles@uk.ibm.com

Wednesday 2nd March 2011



Agenda

- Overview
- Patterns
 - Broker Navigator
 - Patterns Explorer
 - Pattern Parameters
 - Generation
 - Deployment
- Built-in Patterns
- Pattern Authoring
 - Principles and Workflow
 - Design Walk-Through
 - Expressions
 - Enablement
 - Enumerated Types
- Pattern Communities



S H A R E
Technology • Connections • Results

SHARE
in Anaheim
2011

Overview



SHARE
Technology · Connections · Results

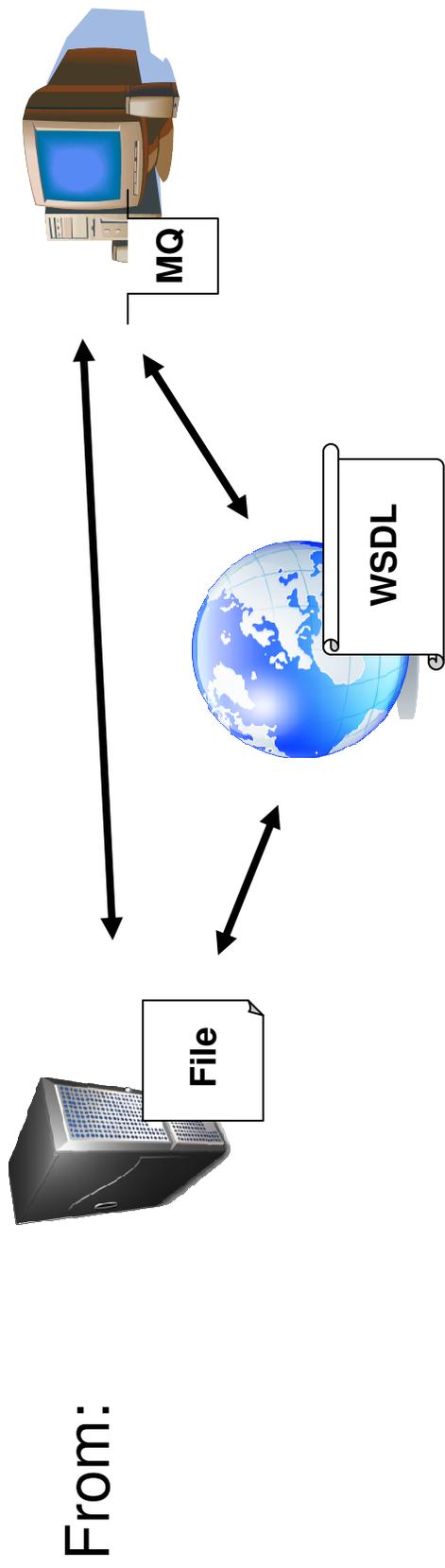
- **Universal Connectivity**
 - Simplify application connectivity to provide a flexible and dynamic infrastructure
 - Routes and transforms messages from anywhere, to anywhere
 - Supports a wide range of protocols
 - MQ, JMS 1.1, HTTP(S), Web Services (SOAP, REST), file, ERP (SAP, SEBL...), TCP/IP
 - Supports a broad range of data formats
 - Binary (C/COBOL), XML, CSV, Industry (SWIFT, EDI, HL7...), IDOCs, user defined
- **Interactions and Operations**
 - Route, filter, transform, enrich, monitor, distribute, decompose, correlate, detect...
- **Simple programming**
 - Patterns based for top-down, parameterized connectivity of common use cases
 - Web Service façades, message oriented processing, queue to file...
 - Construction based for bottom-up assembly of bespoke connectivity logic
 - Message flows to describe application connectivity comprising...
 - Message nodes which encapsulate required integration logic which operate on...
 - Message tree which describes the data in a format independent manner
 - Transformation options include graphical mapping, PHP, Java, ESQ, XSL and WTX
- **Operational Management and Performance**
 - Extensive administration and systems management facilities for developed solutions
 - Wide range of operating system and hardware platforms supported
 - Offers performance of traditional transaction processing environments

SHARE
in Anaheim
2011

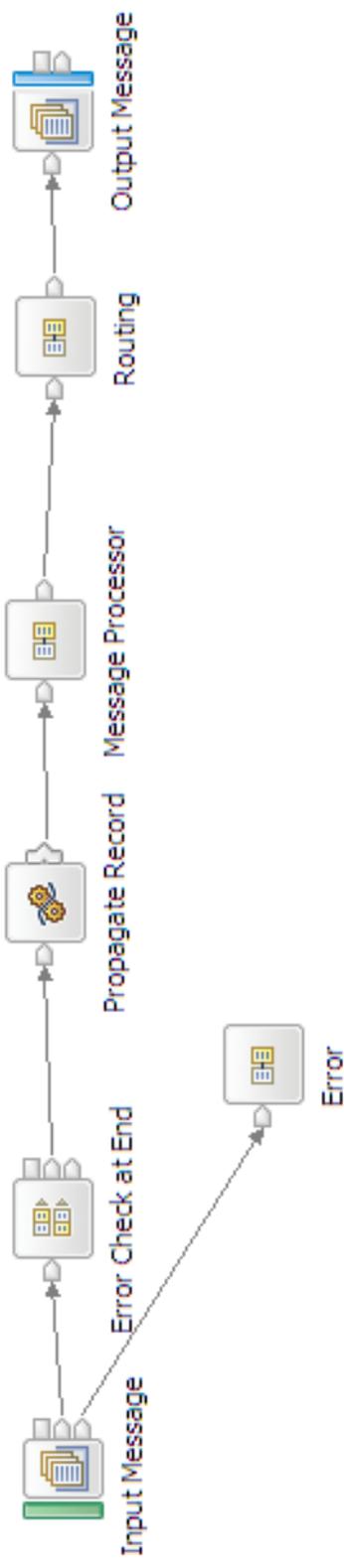
The Challenge



SHARE
Technology • Connections • Results



To:

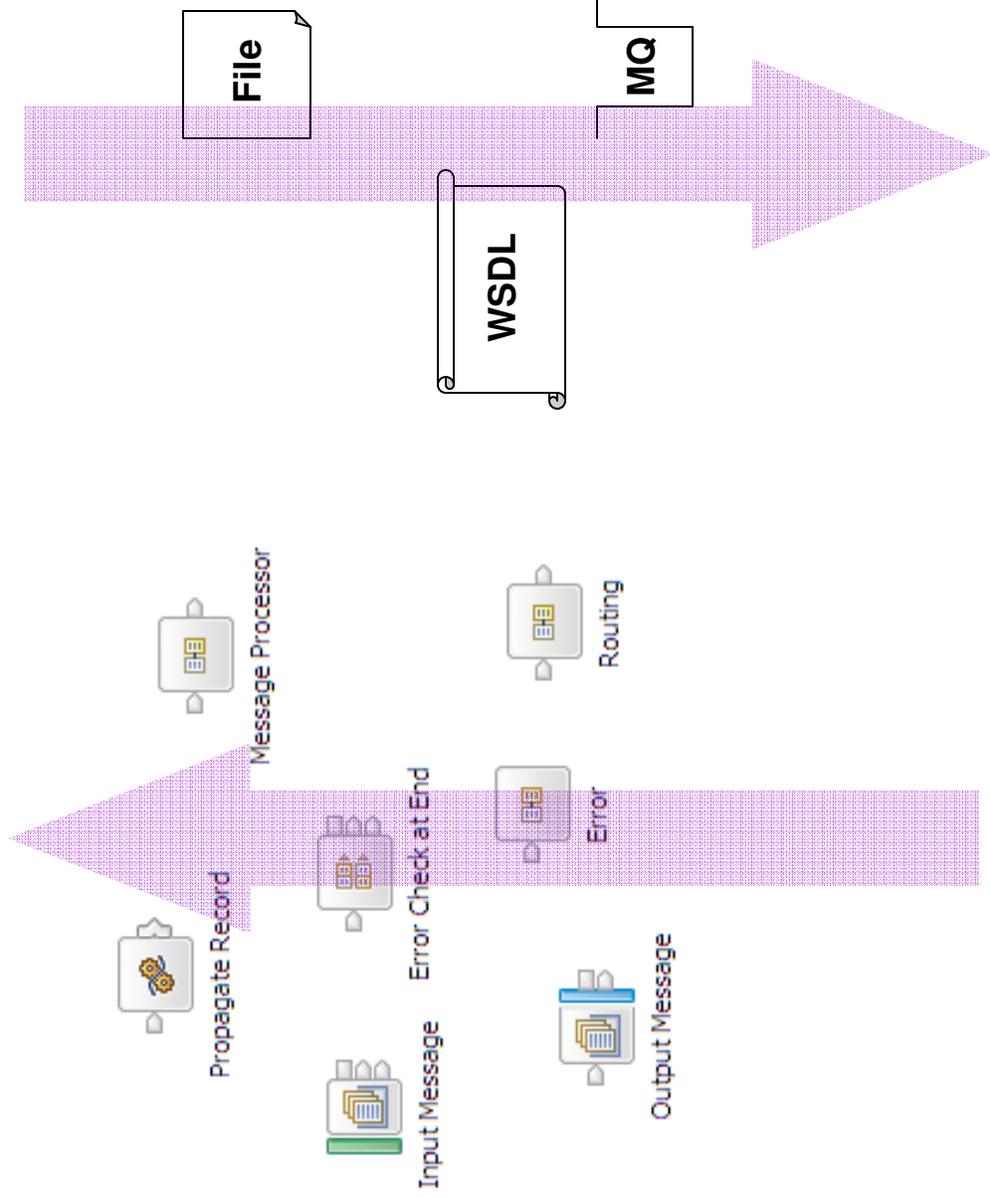


SHARE
in Anaheim
2011

Top-down vs Bottom-up Development



S H A R E
Technology • Connections • Results



SHARE
in Anaheim
2011

Patterns for Simplified Development



S H A R E
Technology · Connections · Results

- Creates top-down, parameterized connectivity solutions
 - Web Service façades, message oriented processing, queue-to-file
- Reduces common problems in flow development
- Communicates best practices to the Broker community
- Complements existing bottom-up construction for bespoke connectivity
- Reduces time-to-value for solution development
- Patterns are a first class citizen in Message Broker
 - Patterns have bubbled right to the top in the navigator view!



S H A R E
Technology • Connections • Results

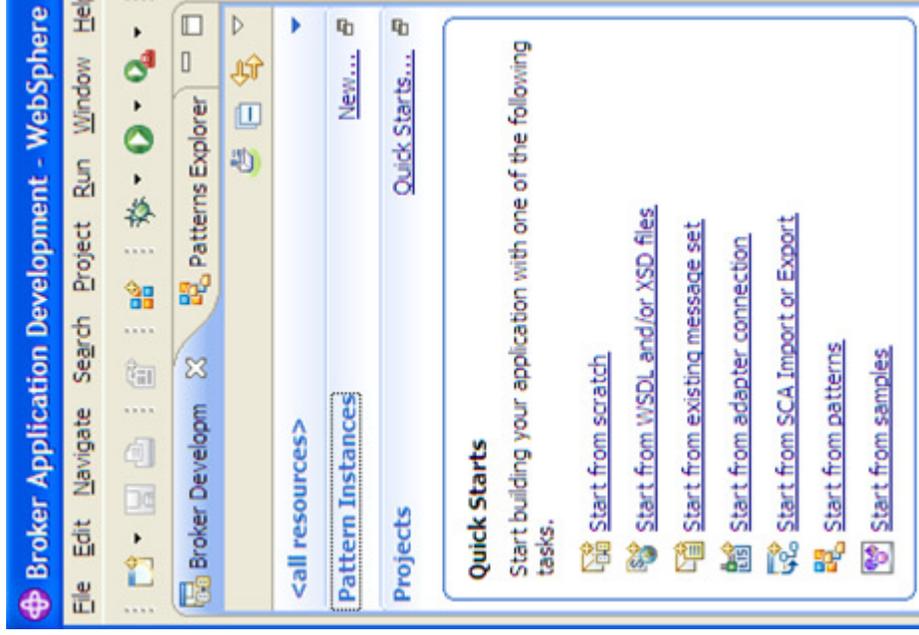
Patterns

SHARE
in Anaheim
2011

Message Broker Navigator



SHARE
Technology · Connections · Results



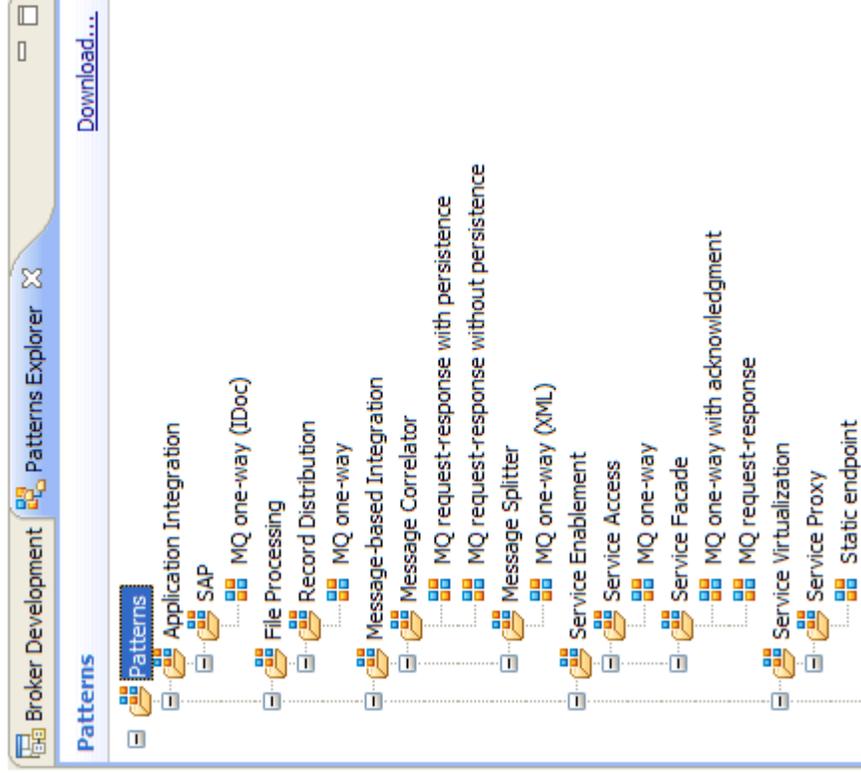
- User is directed towards a new entry point for development
- First class development mechanism
- Augments other starting points:
 - WSDL, SCA, Adapter
- Does not replace the existing bottom up development approach
- Still completely valid to start from flows, message sets etc!

Patterns Explorer



SHARE
Technology · Connections · Results

- Patterns Explorer
 - Pre-supplied IBM patterns
- Pattern categories
 - Groups similar solutions together
 - Message-based integration, service enablement, service virtualization...
- High level help is available
- Describes a class of solutions
 - Leaf nodes are the patterns



Pattern Parameters



SHARE
Technology · Connections · Results

- Customize a pattern using pattern parameters
 - Instance name identifies pattern; duplicate names will optionally be overwritten
 - Parameters are logically grouped into sections
 - Mandatory parameters are indicated via *, missing parameters are indicated via **x**
 - Fields are watermarked and pre-populated, for example: available message sets
 - Detailed help is available for each pattern parameter
 - Click the **Generate** button to create the generated artefacts: message flows, scripts...

A screenshot of the SHARE software interface. The window title is "Input information". It contains several fields: "Routing *" with a dropdown menu showing "No routing"; "Output queue manager" with a text input field; and "Output queue *" with a text input field containing "OUT". There are also checkboxes for "No routing" and "Output queue manager" which are checked. The text "Specify the single output destination" is visible below the "Output queue manager" field.

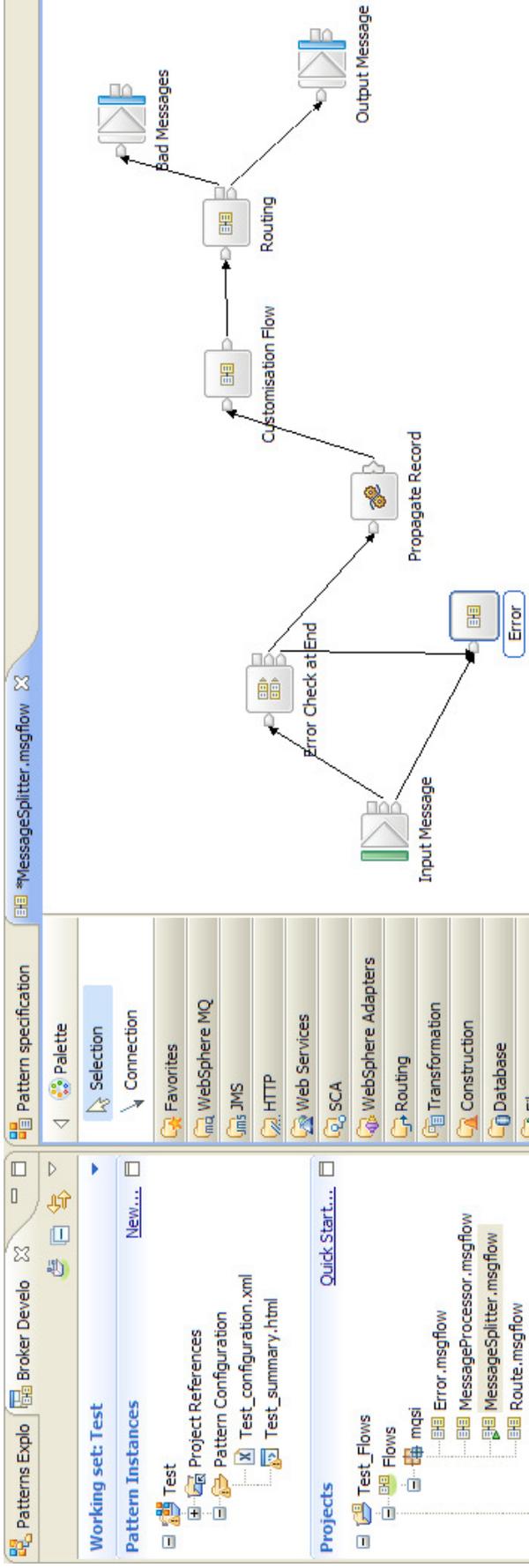
SHARE
in Anaheim
2011

Generation



SHARE
Technology · Connections · Results

- A working set is created, it includes only the current pattern
- The **Projects** view contains the generated projects
- The **Patterns Instances** view contains the pattern instance projects
 - Includes the configuration XML and summary page
- Patterns does not attempt to provide life cycle management!



SHARE
in Anaheim
2011

Deployment



S H A R E
Technology · Connections · Results

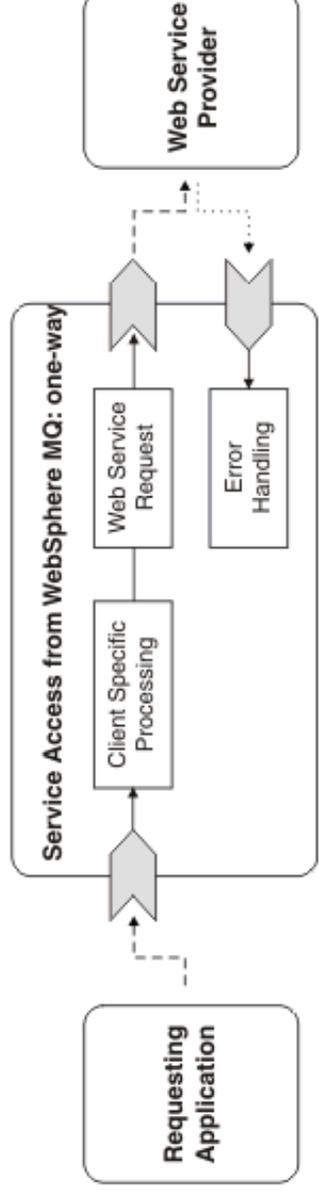
- Nothing changes from regular build and deploy activities
 - Build BAR file as usual from generated assets
 - Deploy as usual, through MBX or the toolkit
- Pattern instance may create additional runtime controls
 - For example, a UDP to control error logging/trace
 - These can be customized as normal

Built-In Patterns



SHARE
Technology · Connections · Results

- Message Broker provides a core set of built-in patterns
- These implement a variety of common scenarios
 - Web service front end to a MQ based application
 - Processing data stored in a file and routing to one or more queues
 - Adding a proxy in front of a web service provider
 - Processing data from an SAP system and routing to MQ
 - Shredding messages and routing to one or more queues
- Patterns are selected based on client feedback and field experience
- This core set of patterns continues to grow with each release
 - MQ to web service pattern added in Message Broker FP1



SHARE
in Anaheim
2011



S H A R E
Technology • Connections • Results

Pattern Authoring

Principles (Part I)



S H A R E
Technology · Connections · Results

- Patterns and pattern authoring are first class concepts
- Pattern authoring is a design activity
 - It may be long lived
 - It is often not sequential
- Using patterns is a top-down activity driven by a requirement, but:
 - Authoring an exemplar is (typically) a bottom-up activity
 - So pattern authoring must bridge these two different approaches
- Patterns have their own development cycle
- We always start with a working exemplar - one or more Broker projects
- Creating the exemplar is part of the pattern authoring process

Principles (Part II)



S H A R E
Technology · Connections · Results

- The Broker implementation of pattern authoring is a Toolkit experience
- Pattern authoring is a managed user experience - a user should not need to understand Broker internals, or other implementation technologies
- A pattern encapsulates one or more exemplar projects and a set of configuration that:
 - Looks down into the exemplar projects to select target properties
 - Looks up to the user interface that will be presented to the pattern user
- Provide guidance so that a pattern author can create exemplars ready for pattern authoring



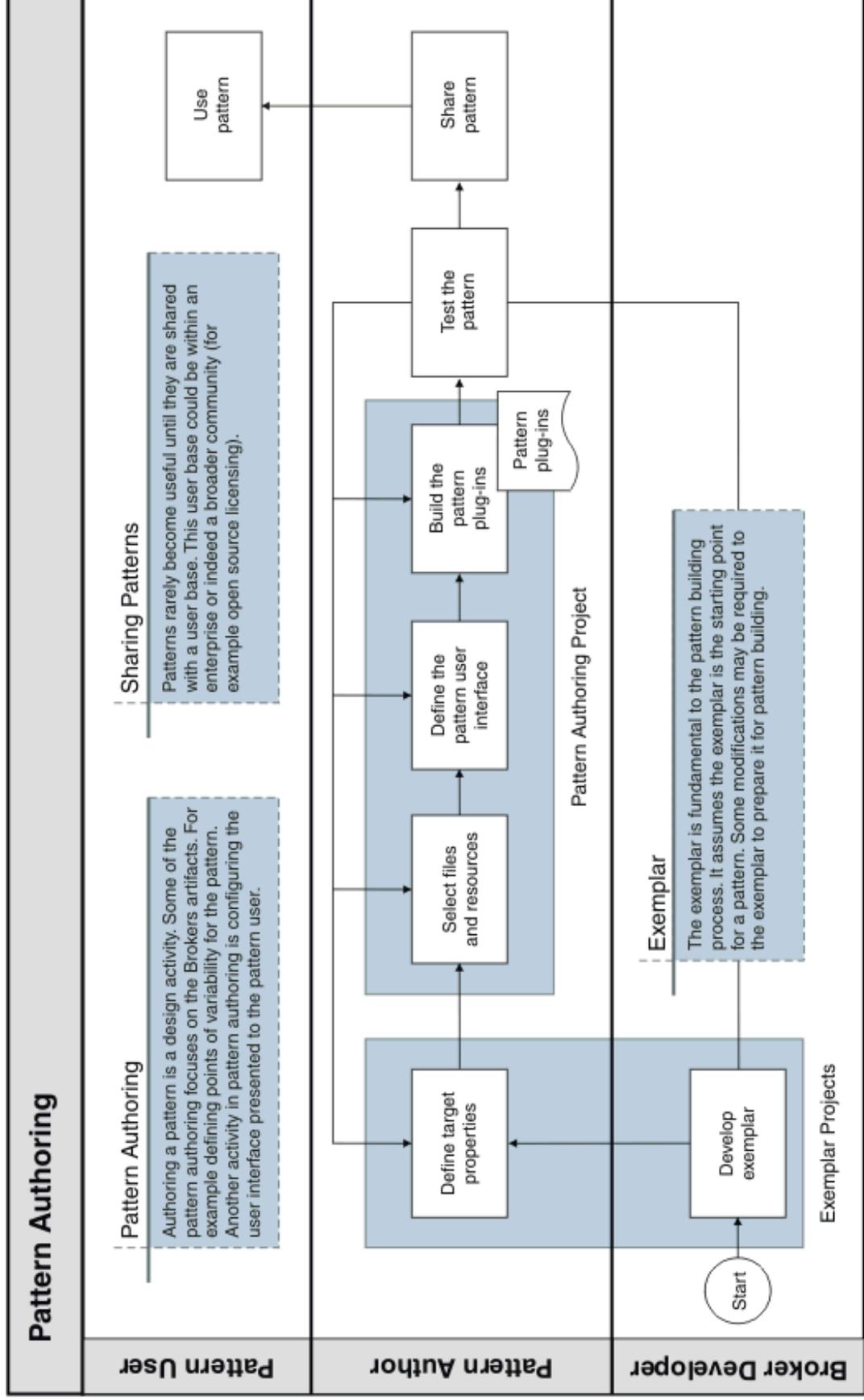
S H A R E
Technology • Connections • Results

Workflow

Workflow



SHARE
Technology · Connections · Results





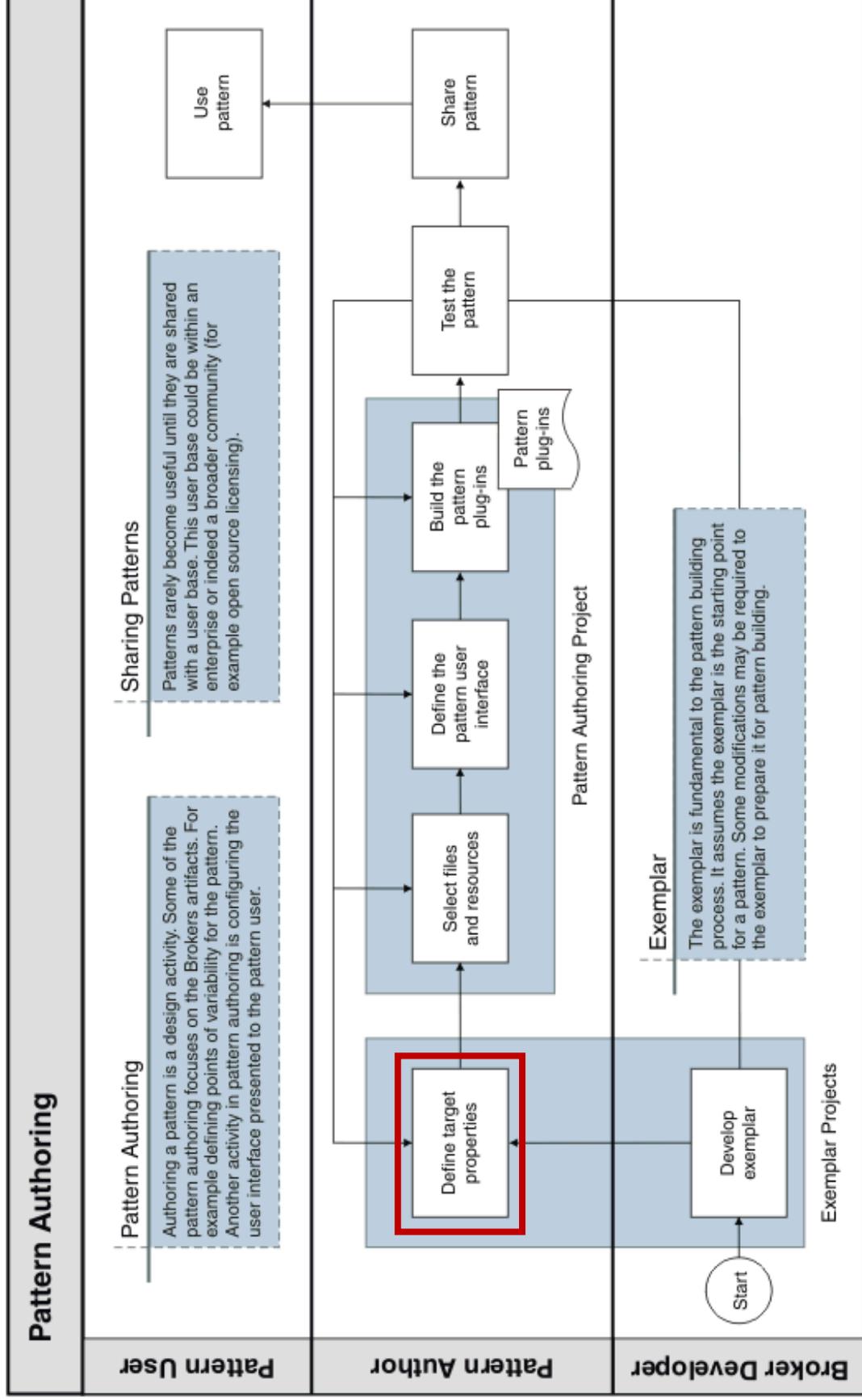
S H A R E
Technology • Connections • Results

Design Walk Through

Define the Target Properties



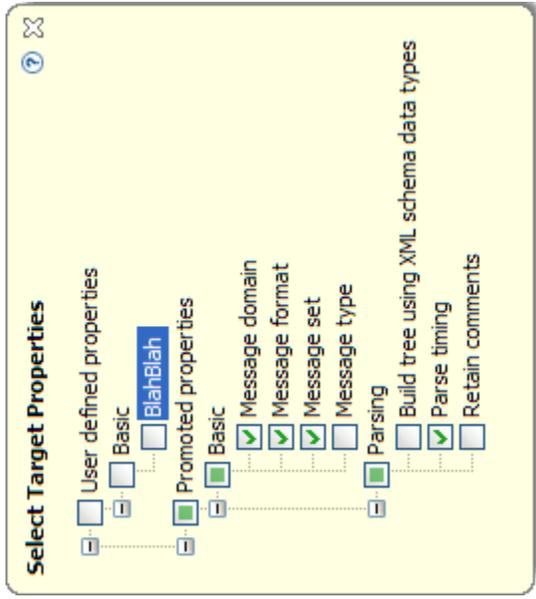
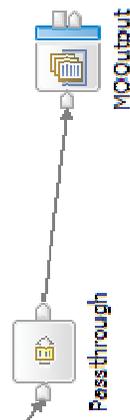
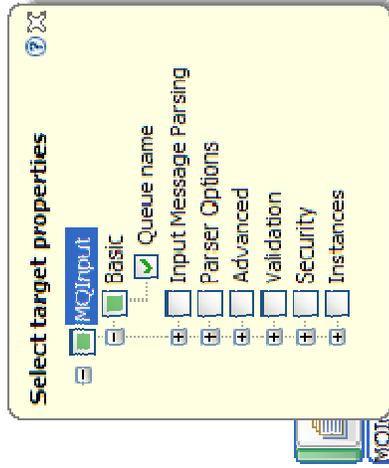
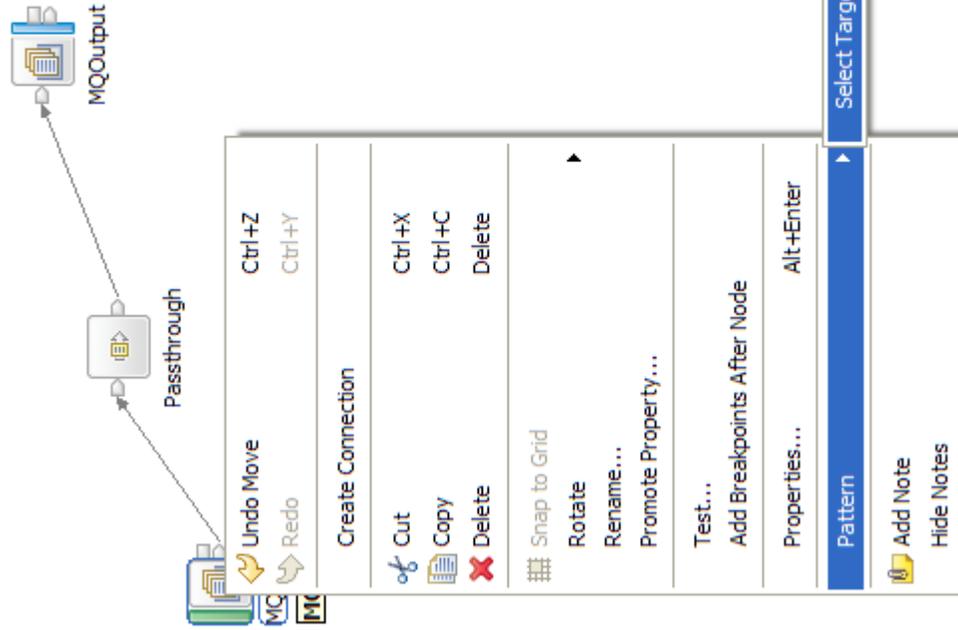
SHARE
Technology · Connections · Results



Define the Target Properties



SHARE
Technology · Connections · Results



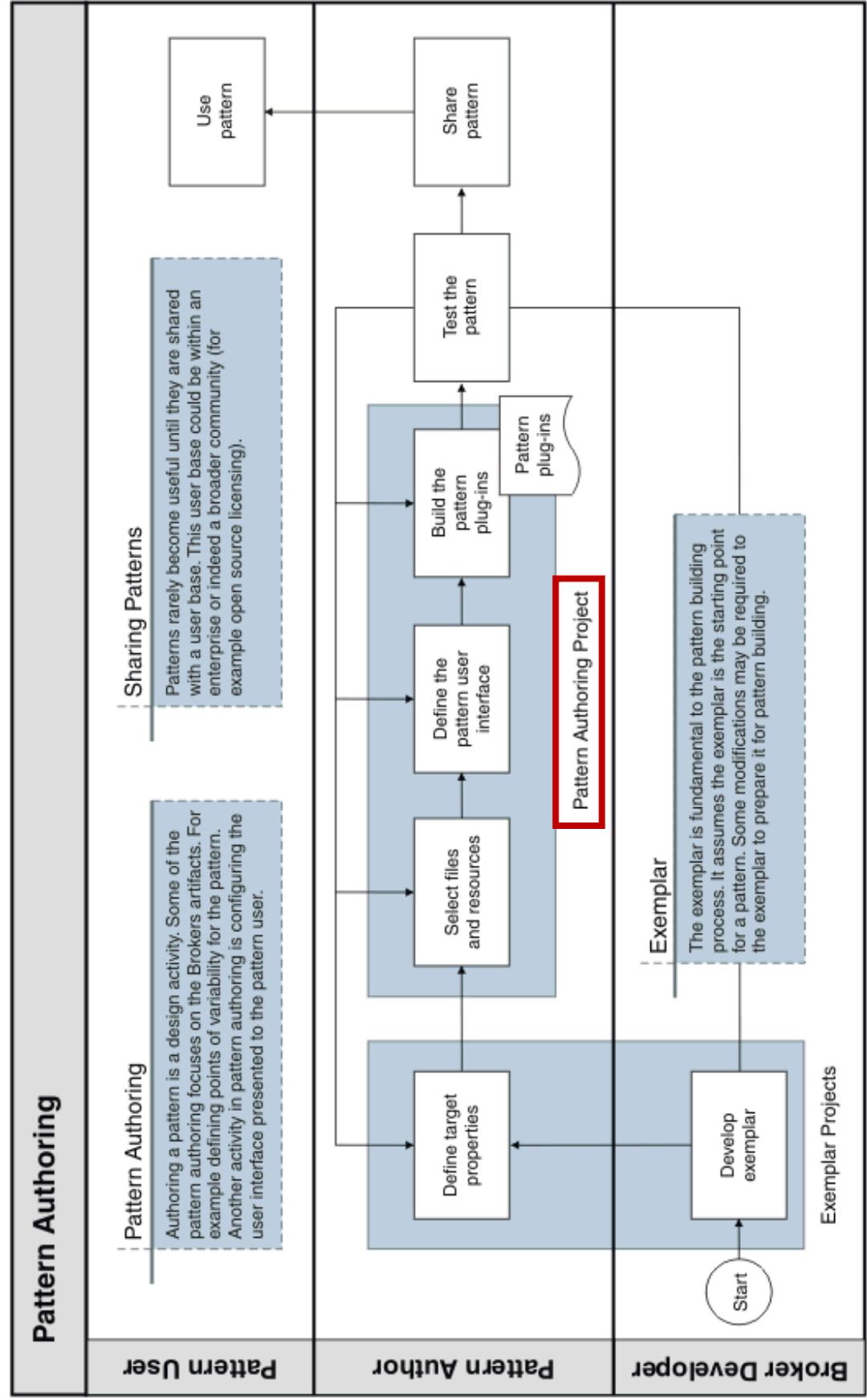
Design Points



S H A R E
Technology • Connections • Results

- Patterns are created in the Broker Toolkit
- Patterns are designed in a Pattern Authoring project and editor
- Variability is expressed in the Toolkit editors
 - The first release supports property variability in the Flow Editor
 - Property variability is based on flow and User Defined Properties (UDPs)
 - Adding variability in the editors also supports a use case where exemplar projects are used in multiple patterns

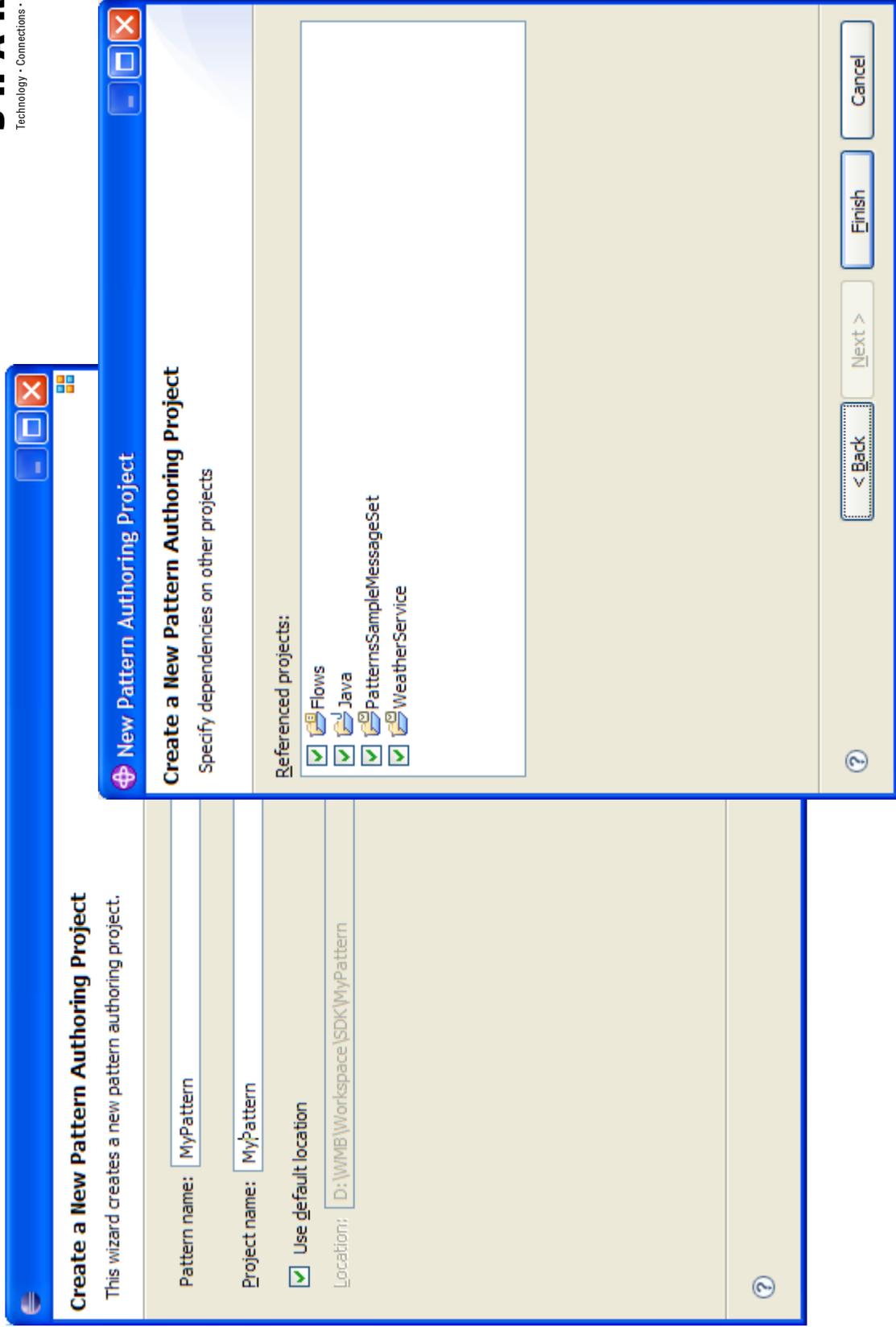
Create A Pattern Authoring Project



Create A Pattern Authoring Project



S H A R E
Technology · Connections · Results



Design Points



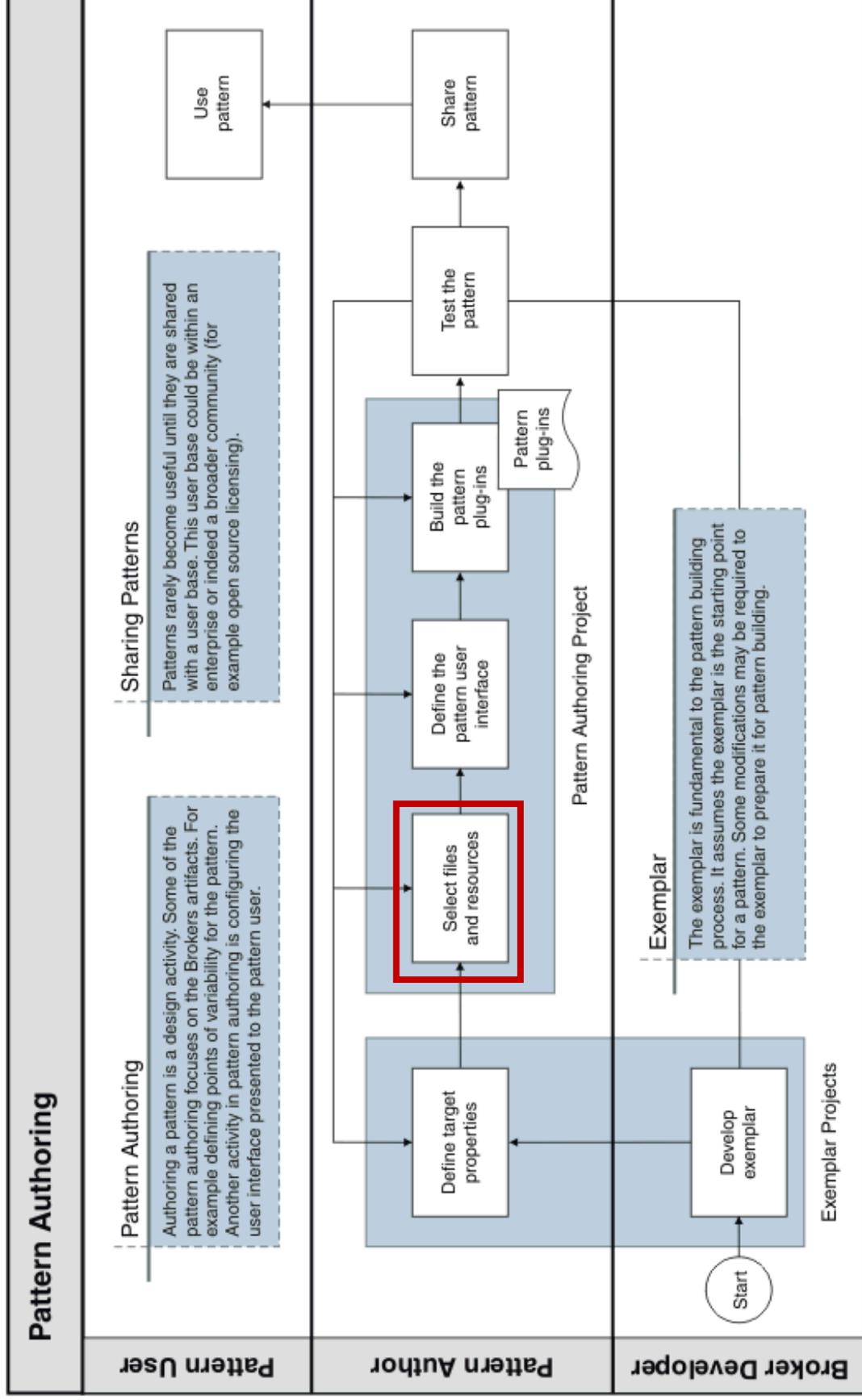
S H A R E
Technology · Connections · Results

- A pattern can generate one or more message flow projects
- A pattern can contain references to projects which the end user is expected to have in their workspace
- A pattern generates projects that match the projects in the exemplars
 - Projects are named following pattern instance naming conventions. For example, if the exemplar project name is `invoicing` and the pattern instance name is `foo`, then the generated project will be called `foo_invoicing`. This simple rule favours convention over configuration. The pattern author can name their exemplar projects accordingly.
- The pattern author selects their exemplar projects when a new Pattern Authoring project is created
 - The Pattern Authoring project maintains references to the exemplar projects
 - These references are standard Eclipse project dependencies
 - The exemplar project files are copied into the pattern plug-in when the pattern is built

Select The Files and Resources



SHARE
Technology · Connections · Results



Select The Files and Resources



SHARE
Technology · Connections · Results

Source Files

Select the source files to include in your pattern. You can also view the target properties available in those source files here.

Select Source Files

type filter text

The screenshot shows a tree view of a project structure. The root folder is 'Flows', which contains a sub-folder 'mqsi'. Under 'mqsi', there are several files: 'Request.esql', 'Request.msgflow', 'Response.esql', and 'Response.msgflow'. The 'Response.esql' and 'Response.msgflow' files have a green checkmark icon next to them, indicating they are selected. Below the 'mqsi' folder, there are other project folders and files: '.project', 'Java', 'bin', 'JavaNode.class', 'src', 'JavaNode.java', '.classpath', '.project', 'WeatherService', 'Weather', 'com', 'cdyne', 'wis', 'weatherws', 'Weather.mxsd', 'Weather.wsdl', 'Weather_InlineSchema.mxsd', 'org', 'xmlsoap', 'schemas', and 'soap'. The 'org', 'xmlsoap', 'schemas', and 'soap' folders also have green checkmarks next to them.

Target Properties

type filter text

The screenshot shows a list of target properties. The list contains seven entries, each with a small icon to its left: 'Flows.mqsi.Request.MQInput.messageDomainProperty', 'Flows.mqsi.Request.MQInput.queueName', 'Flows.mqsi.Request.MQInput.validateTiming', 'Flows.mqsi.Request.PromotedProperty.messageDomainProperty', 'Flows.mqsi.Request.PromotedProperty.messageFormatProperty', 'Flows.mqsi.Request.PromotedProperty.messageSetProperty', and 'Flows.mqsi.Request.PromotedProperty.validateTiming'.

Change Project References

Refresh

Design Points



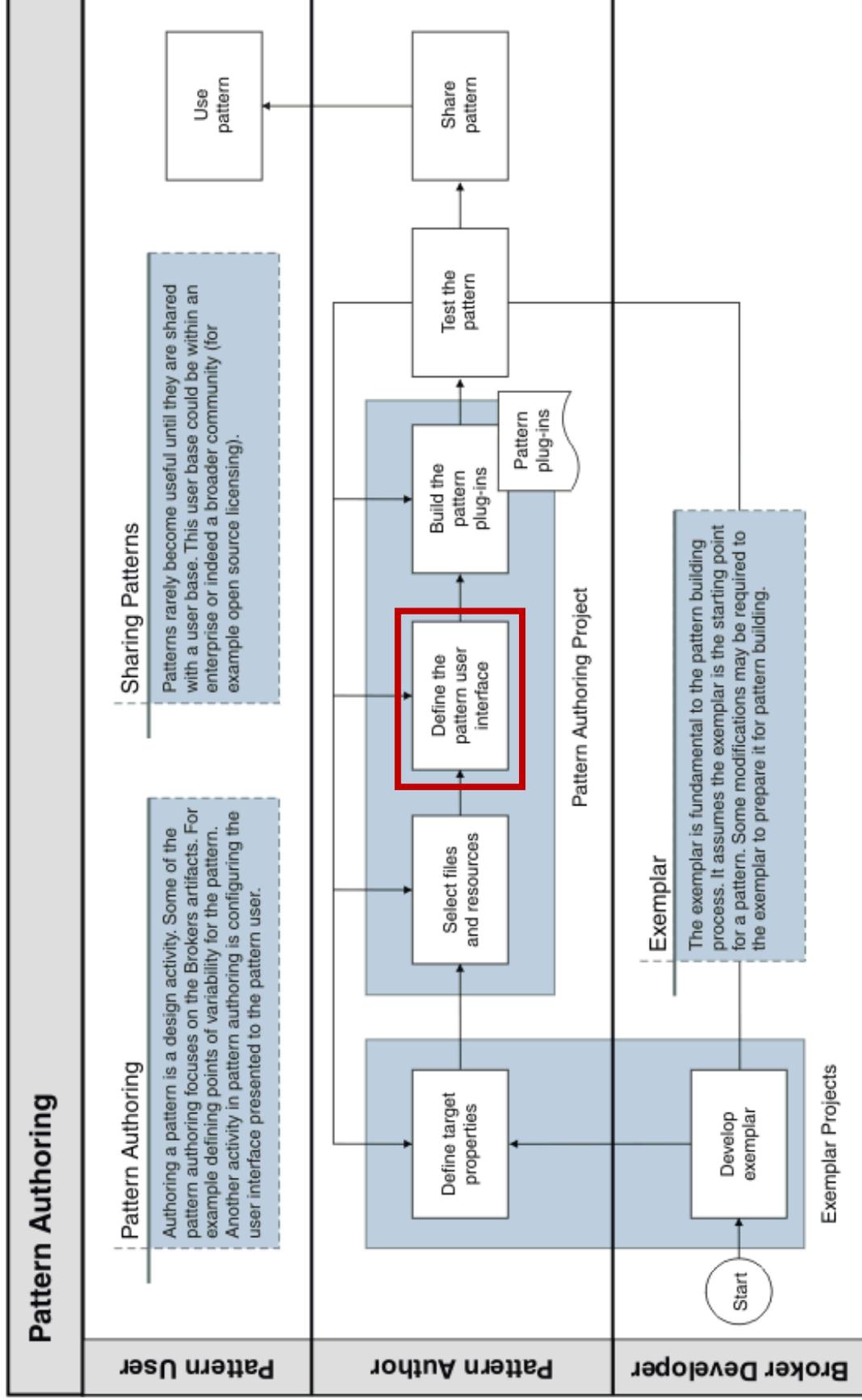
S H A R E
Technology · Connections · Results

- A pattern author chooses the files to include from their exemplar projects
 - This supports a use case where a pattern author has a library project that contain many re-usable assets not all of which are applicable to any given pattern.
- Choosing a file implicitly selects all target properties in that file
 - By default, all files in an exemplar project are selected
- Target properties can be added and removed in the exemplar projects
 - The Pattern Authoring project can be refreshed to pick these up
 - Double clicking a file opens the editor
 - Authoring a pattern is a non sequential design activity!
- Depending on the project type, some files and directories may need to be excluded (such as the `bin` directory in a Java project)

Define The User Interface



SHARE
Technology · Connections · Results



Configure Pattern Parameters (Before)



SHARE
Technology · Connections · Results

Pattern Parameters

Configure your pattern parameters and groups. Associate pattern parameters with target properties.

Groups and Parameters

Default Group

- Message Format Property (MQInput)
- Message Set Property (MQInput)
- Message Type Property (MQInput)
- Blah Blah (UserDefinedProperty)
- Queue Name (MQInput)
- Message Domain Property (MQInput)

Buttons: Add Group..., Add Parameter..., Edit..., Delete, Enumerated Types...



Edit Parameter Logging

Configure the Pattern Parameter

Configure the pattern parameter and how it is displayed to pattern users.

Basic | Expression

Parameter Display

Display name:

Default value:

Parameter editor:

Enumerated type:

Parameter Options

- Hide the parameter
Select this option to make the parameter hidden and use an XPath expression to set its value when a pattern instance is created.
- Translate the default value
Select this option if you want to put the default value for this parameter into a Java properties file ready for translation. Leave this option disabled for single language patterns.
- Configure during deployment
Select this option if you want to generate help text that indicates whether the parameter is intended to be configured in the BAR file.
- Mandatory parameter
Select this option to generate help text that shows the pattern user has to enter a value for this parameter. Mandatory parameters also display a watermark to help guide the pattern user.

Watermark:

Help Text (HTML):

Enumerated Types...

OK Cancel

Configure Pattern Parameters (After)



S H A R E
Technology · Connections · Results

Pattern Parameters

Configure your pattern parameters and groups. Associate pattern parameters with target properties.

Groups and Parameters

A screenshot of a software configuration window titled "Groups and Parameters". The window contains a tree view on the left with the following items: "General", "Logging", "Queue name", "Input Information" (highlighted with a blue selection bar), "Message set", "Message domain", "Message type", and "Message format". Each item has a small icon to its left. To the right of the tree view is a large empty text area. At the bottom of the window, there are four buttons: "Add Group...", "Add Parameter...", "Edit...", and "Delete". On the far right, there are two small icons: a yellow arrow pointing up and a yellow arrow pointing down, and a button labeled "Enumerated Types...".

Design Points



S H A R E
Technology · Connections · Results

- Every file containing one or more target properties has a pattern parameter group created automatically
- Likewise, every target property has a pattern parameter created automatically
- The pattern is always in a valid state even with no refinement
- Pattern parameter groups can be added, deleted and edited as required
 - The default name for a pattern parameter group is based on the file name
 - Only empty pattern parameter groups can be deleted
- **Pattern parameters can be added, deleted and edited as required**
 - The default name for a pattern parameter is based on the target property name
 - Only empty pattern parameters can be removed – target properties cannot float free, they must always be mapped to a pattern parameter
- The 1:1 mapping between pattern parameters can be changed by shuffling the target properties around (drag and drop)

Add Pattern Categories

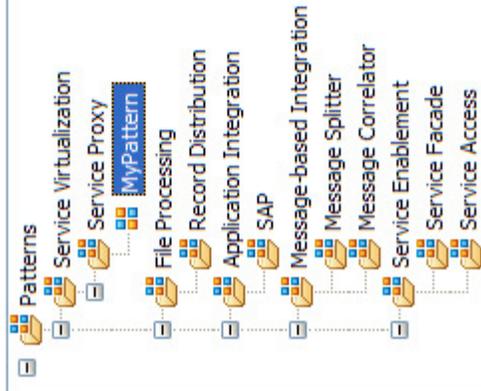


SHARE
Technology • Connections • Results

Categories

Create new categories and assign your pattern to a category.

Categories



Add Category

Remove Category

Edit Category



SHARE
in Anaheim
2011

Design Points



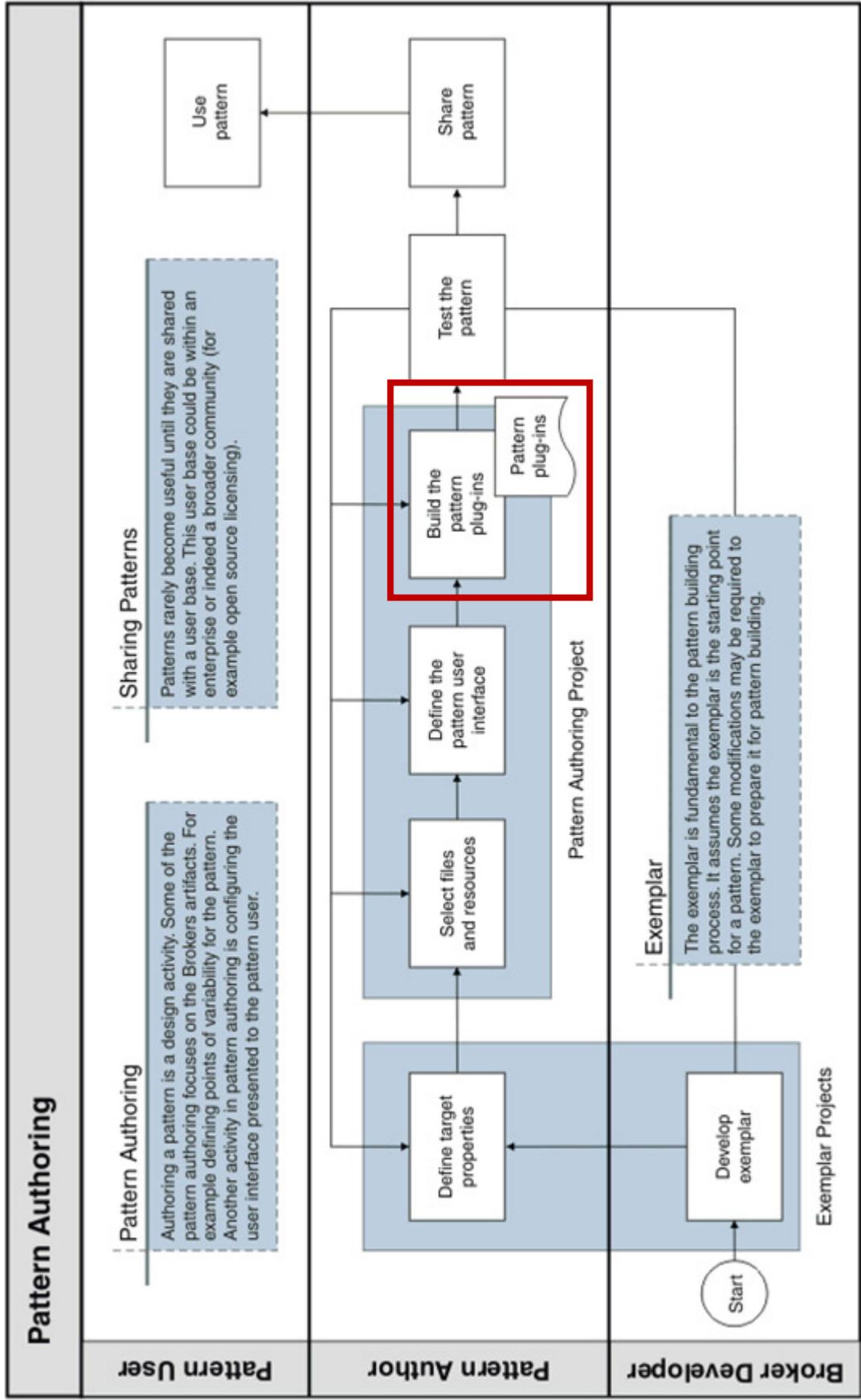
S H A R E
Technology • Connections • Results

- Category specification is typically one HTML file
- Each category gets a directory to store its specification
- Pattern Authoring editor creates a skeleton file in the directory
- Pattern author can use any HTML editor they choose
- Specification is packaged into the pattern plug-ins when they are created
- By convention the category specification is called `overview.htm`
- All files and sub-directories are packaged as well
 - For example, to brand the category with images and stylesheets (CSS)
- There are no restrictions on where new categories are added
 - The pattern can be added to any category either existing or new

Build The Pattern Plug-ins



S H A R E
Technology · Connections · Results



Build The Pattern Plug-ins



SHARE
Technology · Connections · Results

Create Pattern

Test your pattern by configuring your pattern plug-in information, click "Create Pattern Plug-ins", and click Test Pattern".

Plug-in Information

 Configure the unique identifier for your pattern plug-in.

| | | |
|---------------|---|---|
| Pattern name: | MyPattern |  |
| Plug-in ID: | com.your.company.domain.MyPattern |  |
| Version: | 1.0.0.0 |  |
| Provider: | Your Company Name | |
| Description: | Plug-in created by the Pattern Authoring editor | |

Translation Options

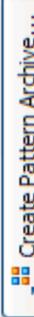
If you enable this option, the Pattern Authoring editor creates two additional NLS plug-ins. These plug-ins are set up so that you can drop in translated resources, such as Java property files. If you are creating a single language pattern, do not select this check box.

Create translation plug-ins (*.n1 and *.doc.n1)

Pattern Distribution

After you have created and tested your pattern plug-ins (see the Plug-in ID above for the plug-in names), package your pattern by clicking "File > Export > General > Archive File" to export these plug-ins.

To use the pattern plug-ins, the pattern user must extract the exported archive files into the default WebSphere Message Broker Toolkit plug-ins directory:



Design Points



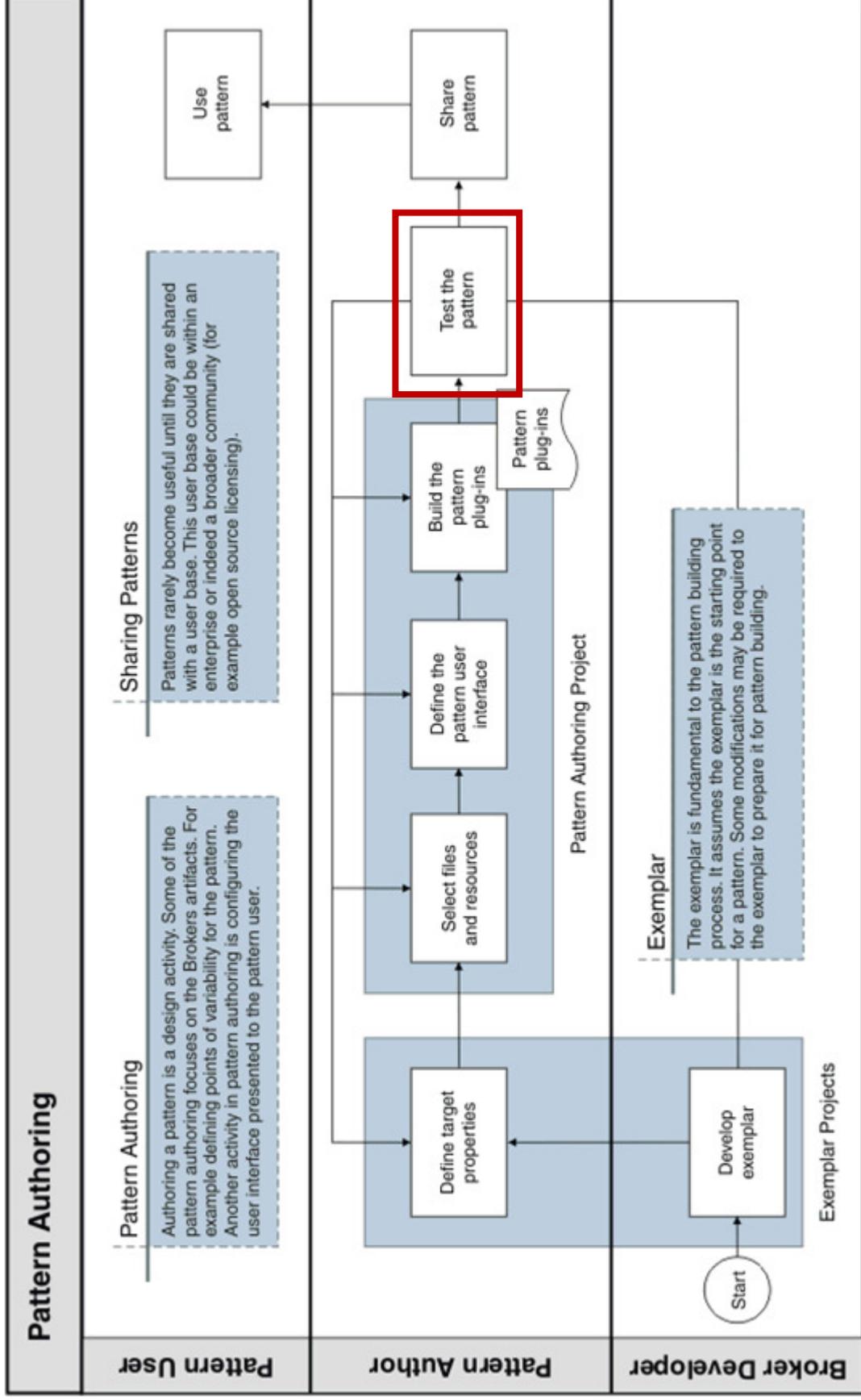
S H A R E
Technology · Connections · Results

- The Pattern Authoring Editor creates Eclipse plug-ins
 - Eclipse provides excellent support for plug-ins and features for distribution
- Some plug-in information is required such as provider and version
- It is not recommended to edit the generated plug-ins
 - One exception is to add translated property files into the NLS plug-ins

Test The Pattern



SHARE
Technology · Connections · Results



Test The Pattern



SHARE
Technology · Connections · Results

The screenshot shows the IBM Broker Development Patterns Explorer interface. The left pane displays a tree view of patterns, with 'My Pattern' selected. The right pane shows the details for 'My Pattern', including a 'View Pattern Specification' section with instructions, a 'My Pattern' title, a 'Solution' section, and a 'Related tasks' section with several links.

Patterns Explorer

- Patterns
 - Application Integration
 - SAP
 - MQ one-way (IDoc)
 - File Processing
 - Record Distribution
 - MQ one-way
 - Message-based Integration
 - Message Correlator
 - MQ request-response with persistence
 - MQ request-response without persistence
 - Message Splitter
 - MQ one-way (XML)
 - My Pattern**
 - Service Enablement
 - Client Access
 - MQ One Way Service Access
 - MQ Request Response Service Access
 - Service Facade
 - MQ one-way with acknowledgment
 - MQ request-response
 - Service Virtualization
 - Service Proxy
 - Static endpoint

Pattern specification

View Pattern Specification

View information about the selected pattern and click the "Create New Instance" button or click [here](#) to start using a pattern.

My Pattern

Solution

Related tasks

Read the following section for information about how to apply and use this pattern.

- [Constraints on the use of the pattern](#)
- [Tasks to complete before applying the pattern](#)
- [Apply and configure the pattern](#)
- [Parameters for the pattern](#)
- [Tasks to complete after generating the pattern](#)

Design Points



S H A R E
Technology · Connections · Results

- Testing a pattern requires a re-launch of the Toolkit
 - Run configurations are an often misunderstood area of Eclipse development so the Pattern Authoring Editor provides a simple way to re-launch the toolkit:



- The pattern appears in the Pattern Explorer and is ready to test
 - The Pattern Authoring Editor will allow new categories to be created
 - A skeleton pattern specification is created which can also be changed
- This step completes the application development cycle for a pattern
 - In practice a pattern author would loop around these steps many times!



S H A R E
Technology • Connections • Results

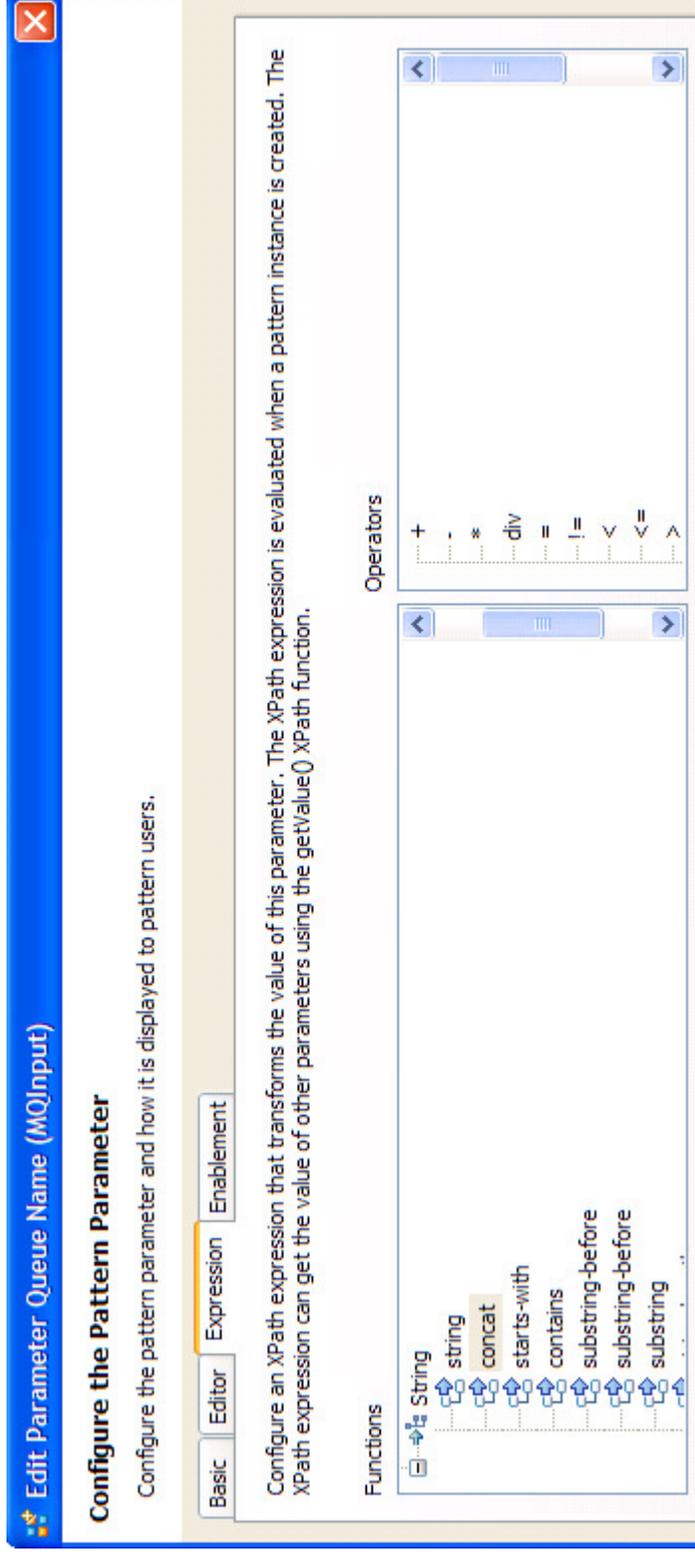
Advanced Topics

Transformation



SHARE
Technology - Connections - Results

- The Pattern Authoring editor uses XPath as its expression language
 - XPath is a general purpose expression language!
- XPath expressions can transform pattern parameter values:
 - Pattern authors can configure an XPath expression for a pattern parameter
 - Expressions are evaluated when the pattern instance is generated

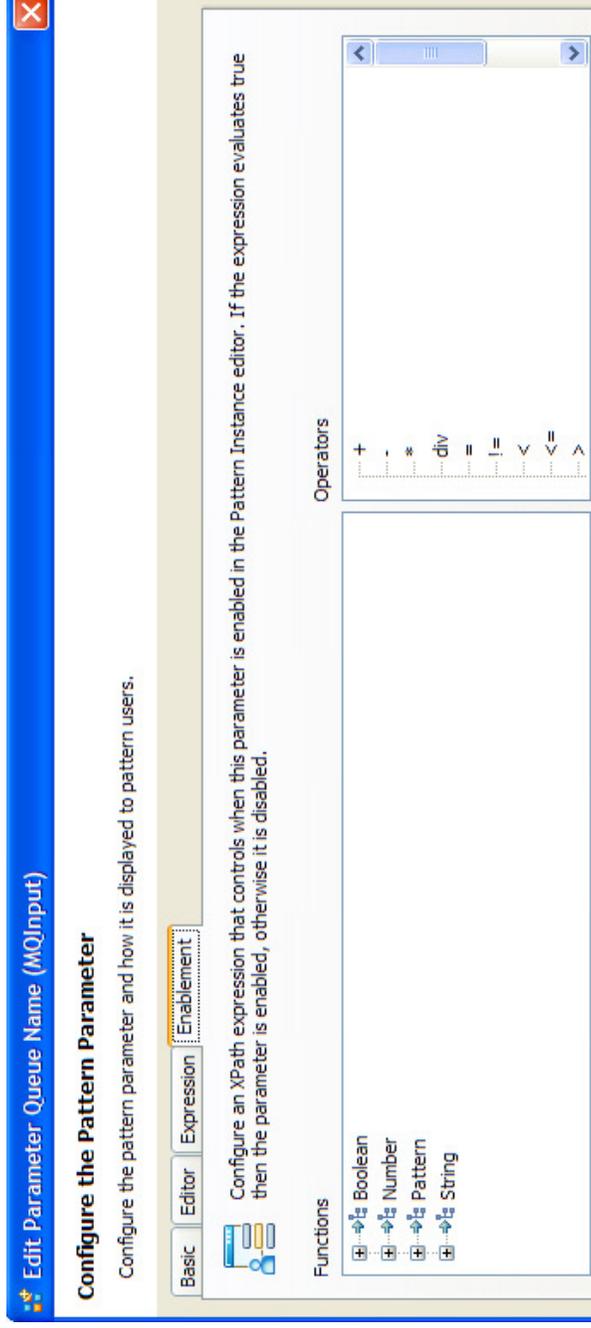


Enablement



SHARE
Technology • Connections • Results

- Enablement uses an XPath expression to control when a pattern parameter is enabled in the Pattern Instance editor



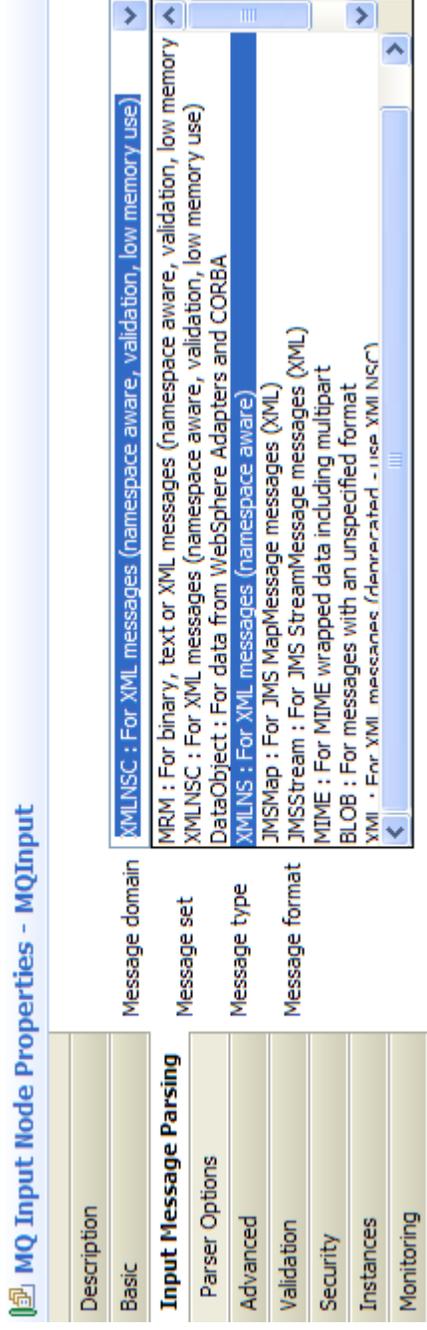
- The enablement expression is evaluated every time a pattern parameter referenced in the XPath expression changes value
- If the result of the evaluation is true then the editor is enabled

SHARE
in Anaheim
2011

Enumerated Types



- Target properties are strongly typed
 - String, integer, boolean and enumerations and the most common types
 - An enumeration is a list of permissible values for a given property:



- The Pattern Authoring editor has full support for enumerations
 - An enumerated type is automatically created when a target property is added
 - The enumerated type includes the display names and property values
 - The list of values presented to the pattern user can be reduced if required



S H A R E
Technology • Connections • Results

Pattern Communities

Pattern Communities



S H A R E
Technology · Connections · Results

- Pattern authoring creates patterns whose value multiplies as they are shared and used by a community of developers
- Public and private communities are equally interesting!
- So how to distribute and share patterns effectively?
- A community needs to offer more than just a repository of assets!
 - Space for content such as patterns and subflow nodes
 - Forums to host discussions and threaded conversations
 - Categorisation is important as the community grows! (taxonomies)
 - Essential administrative functions such as user management
 - News articles and broadcasts keep the site fresh
- There are literally dozens of content management systems available
- On the commercial side we have tried [Lotus Connections](#) and [Rational Asset Manager](#)
- Open source projects such as [Drupal](#) and [Joomla](#) are compelling!

SHARE
in Anaheim
2011

Packaging a Pattern



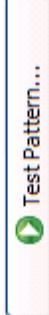
S H A R E
ms - Results

Create Pattern

Test your pattern by configuring your pattern plug-in information, click "Create Pattern Plug-ins", and click Test Pattern*.

Plug-in Information

 Configure the unique identifier for your pattern plug-in.

| | | |
|---------------|--|---|
| Pattern name: | <input type="text" value="MyPattern"/> |  |
| Plug-in ID: | <input type="text" value="com.your.company.domain.MyPattern"/> |  |
| Version: | <input type="text" value="1.0.0.0"/> |  |
| Provider: | <input type="text" value="Your Company Name"/> | |
| Description: | <input type="text" value="Plug-in created by the Pattern Authoring editor"/> | |

Translation Options

If you enable this option, the Pattern Authoring editor creates two additional NLS plug-ins. These plug-ins are set up so that you can drop in translated resources, such as Java property files. If you are creating a single language pattern, do not select this check box.

Create translation plug-ins (*.nl1 and *.doc.nl1)

Pattern Distribution

After you have created and tested your pattern plug-ins (see the Plug-in ID above for the plug-in names), package your pattern by clicking "File > Export > General > Archive File*" to export these plug-ins.

To use the pattern plug-ins, the pattern user must extract the exported archive files into the default WebSphere Message Broker Toolkit plug-ins directory:



Pattern Archives



S H A R E
Technology · Connections · Results



- The goal is to simplify the packaging and distribution of patterns
- Creating Eclipse update sites and features is a fairly complex task
- A pattern archive contains the pattern plug-ins and an Eclipse feature
 - The version of the feature matches the pattern version:

Plug-in Information

 Configure the unique identifier for your pattern plug-in.

| | | |
|---------------|---|--|
| Pattern name: | MyPattern |  Create Pattern Plug-ins |
| Plug-in ID: | com.your.company.domain.MyPattern |  Test Pattern... |
| Version: | 1.0.0.0 |  Debug Pattern... |
| Provider: | Your Company Name | |
| Description: | Plug-in created by the Pattern Authoring editor | |

Creating a Community Space



S H A R E



WebSphere Message Broker

Developing patterns and subflow nodes in Message Broker

New forum topics

- Question about the Service Proxy pattern [more](#)

Active forum topics

- Question about the Service Proxy pattern [more](#)

admin

- My account
- ▶ Create content
- ▶ Administrator
- Log out



Search

Search this site:

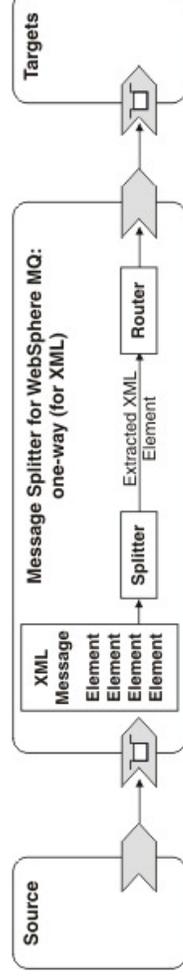
Patterns Community

This site shows you how to build a community web site to share Message Broker patterns, subflow nodes and much more!

The web site is built using Drupal, a very popular and well established open source content management system (CMS):

- Voting is provided by **FiveStar** and **VotingAPI** modules
- Site search is configured (this is a built in module)
- An easy to use **Administration** module is available
- Pages based around an image are provided by the **Image** module
- Categorisation of content made available by the **Taxonomy** module

Message Splitter Pattern



This is the first version of a new message splitter pattern.

Use the Message Splitter for WebSphere MQ: one-way (for XML) pattern to split a large XML message into smaller elements for processing by one or more targets, by using transactional flows and persistent WebSphere MQ messages.

Who's online

There are currently **1 user** and **0** guests online.

Online users

- admin

Syndicate



Who's new

- admin

Taxonomy

- ▼ Patterns
 - Application Integration
 - Database Integration
 - Java
 - Message Based Integration
 - Message Splitter
 - PHP
 - Service Facade

A Shared Repository for Patterns



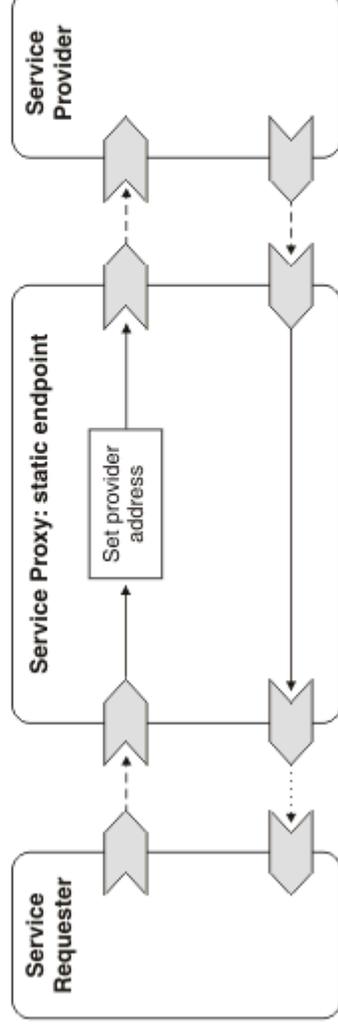
S H A R E
Technology · Connections · Results

Service Proxy Pattern

Use this pattern to provide decoupling between Web service requesters and Web service providers by routing through a virtual service that is bound directly to the target service provider.

Use this pattern:

1. When you do not want the client to access the service directly, either because the address of the endpoint at which the service is located might change, or because it must be hidden to control access to the service
2. To support service access over HTTP or HTTPS
3. When you want to provide logging of all requests to the service provider, or when you want to provide additional error handling capabilities without making changes to the service provider



Average:



Your rating: 2.5 Average: 2.5 (1 vote)

Attachment

Size

MyPattern.patternzip 51.36 KB

SHARE
in Anaheim
2011

Drupal

Average:



Your rating: 2.5 Average: 2.5 (1 vote)



S H A R E

Technology · Connections · Results

- The demonstration system is a standard Drupal v6 install
 - Drupal is a PHP-based open-source content management system (CMS)
 - A key strength of Drupal is its vibrant community of users and developers
 - Drupal has an extensive list of many hundreds of add-on modules
- The following contributed modules were installed:
 - Voting is provided by [FiveStar](#) and [VotingAPI](#) modules
 - Site search is configured (this is a built in module)
 - An easy to use [Administration](#) module is available
 - Pages based around an image are provided by the [Image](#) module
 - Categorisation of content made available by the [Taxonomy](#) module
- It took just over one hour to complete the system set up!

| Search |
|--|
| Search this site: <input type="text"/> |
| <input type="button" value="Search"/> |

| Taxonomy |
|---|
| ▼ Patterns <ul style="list-style-type: none">• Application Integration• Database Integration• Java• Message Based Integration• Message Splitter |

SHARE
in Anaheim
2011

Community Branding



SHARE
Technology · Connections · Results

New forum topics

Question about the Service Proxy pattern

[more](#)

admin

[My account](#)
[Create content](#)
[Administer](#)
[Log out](#)

Taxonomy

► [Patterns](#)
[Application Integration](#)
[Database Integration](#)
[Java](#)
[Message Based Integration](#)
[Message Splitter](#)
[PHP](#)
[Service Facade](#)
[Service Proxy](#)
[User-Defined Subflow Nodes](#)
[Web Services](#)
[WebSphere MQ](#)

Patterns Community

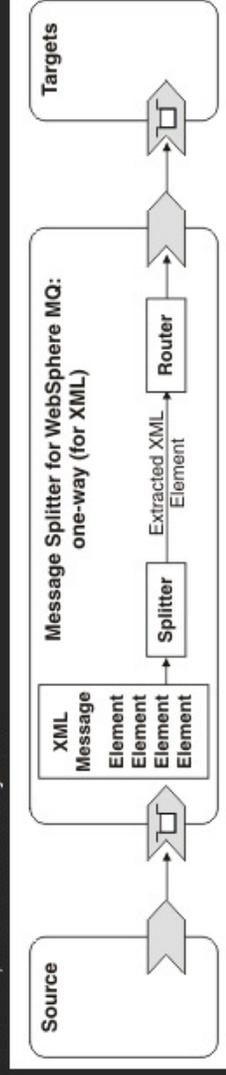
Posted Sun, 09/26/2010 - 15:26 by [admin](#)

This site shows you how to build a community web site to share Message Broker patterns, subflow nodes and much more! The web site is built using Drupal, a very popular and well established open source content management system (CMS):

- Voting is provided by [FiveStar](#) and [VotingAPI](#) modules
- Site search is configured (this is a built in module)
- An easy to use [Administration](#) module is available
- Pages based around an image are provided by the [Image](#) module
- Categorisation of content made available by the [Taxonomy](#) module

Message Splitter Pattern

Posted Sun, 09/26/2010 - 14:24 by [admin](#)



This is the first version of a new message splitter pattern.

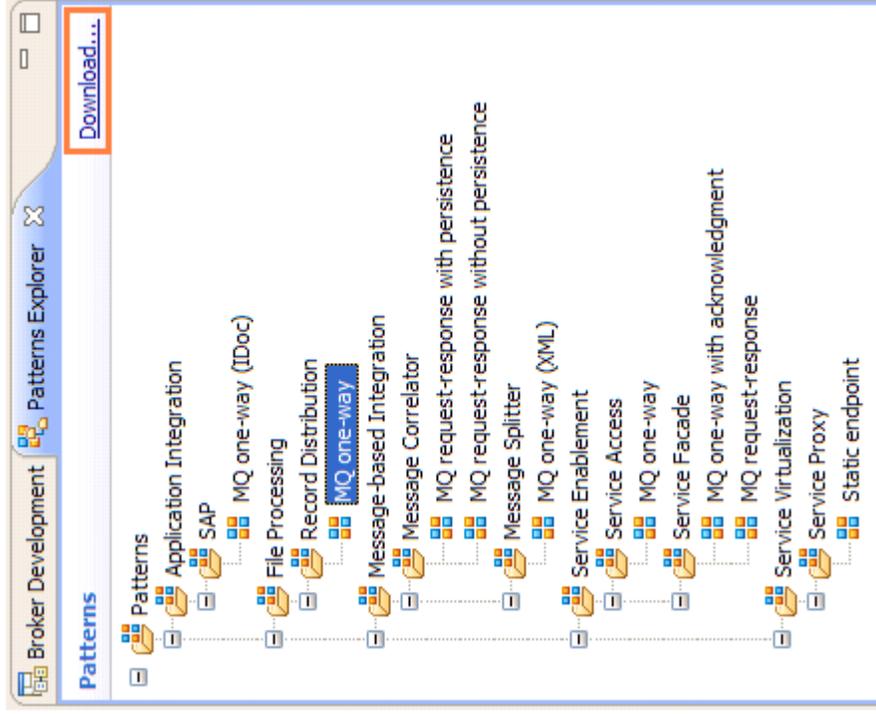
Use the Message Splitter for WebSphere MQ: one-way (for XML) pattern to split a large XML message into smaller elements for processing by one or more targets, by using transactional flows and persistent WebSphere MQ messages.

Use this pattern when you have applications that store information about a number of business transactions and transmit this information to one or more target applications in batches. It can be used to handle large messages without excessive memory use.

Installing Patterns



SHARE
Technology · Connections · Results



- The Pattern Explorer provides a download capability for pattern archives
- Pattern archives can be installed from a local file system or from a URL
- HTTP/S is not supported – instead login, download and install from the file system
- Installation completes without a toolkit restart
- Patterns are uninstalled through the Eclipse updates feature: Help > Software Updates
- .patternzip files can also be double clicked which runs an installer application (the file extension is registered by the toolkit)

SHARE
in Anaheim
2011

Questions – Please Fill Out Your Session Evaluation Form!



S H A R E
Technology • Connections • Results



why is|
why is the sky blue
why is my poop green
why is a raven like a writing desk
why is lil wayne going to jail
why is everyone posting colors on facebook
why is the world going to end in 2012
why is yawning contagious
why is lil wayne in jail
why is my computer so slow

Google Search I'm Feeling Lucky