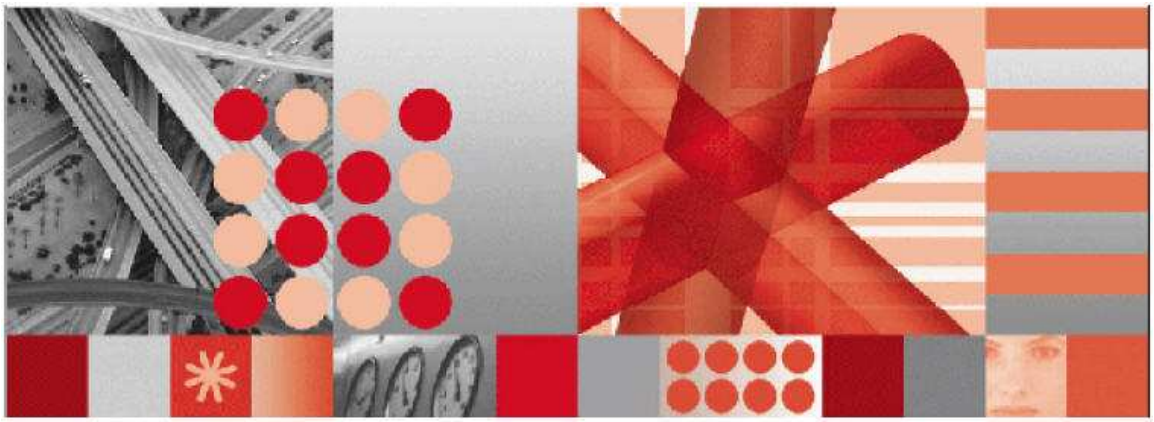
IBM Maximo Asset Management 7.1

IBM Maximo Asset Management for IT 7.1

IBM Tivoli Change and Configuration Management Database 7.1.1

IBM Tivoli Service Request Manager 7.1



Web service interactions

Note

Before using this information and the product it supports, read the information in Notices on page 23

This edition applies to version 7, release 1, modification 1 of IBM Maximo Asset Management, IBM Maximo Asset Management for IT, IBM Tivoli Change and Configuration Management Database, and IBM Tivoli Service Request Manager, and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Web-service based interactions

A new capability has been added to the Integration Framework function in Fix Pack 7.1.1.6. This capability enables practitioners to assemble a web service-based integration very rapidly without the need for authoring code. The capability is based on web services that can be invoked from Maximo Base Services. Existing Integration Framework functions are utilized to achieve this type of integration.

Overview

A web service interaction is a combination of configuration entities in the Maximo Base Services environment that act in concert to achieve the following:

- prepare a request and pass request parameters to the web service
- invoke a web service from a chosen business application
- retrieve the results from the web service in the form of a response
- display the results in the context of the business application
- apply the result data into the application the web service was launched from
- report errors during web service invocation or response processing

Figure 1 illustrates this capability.

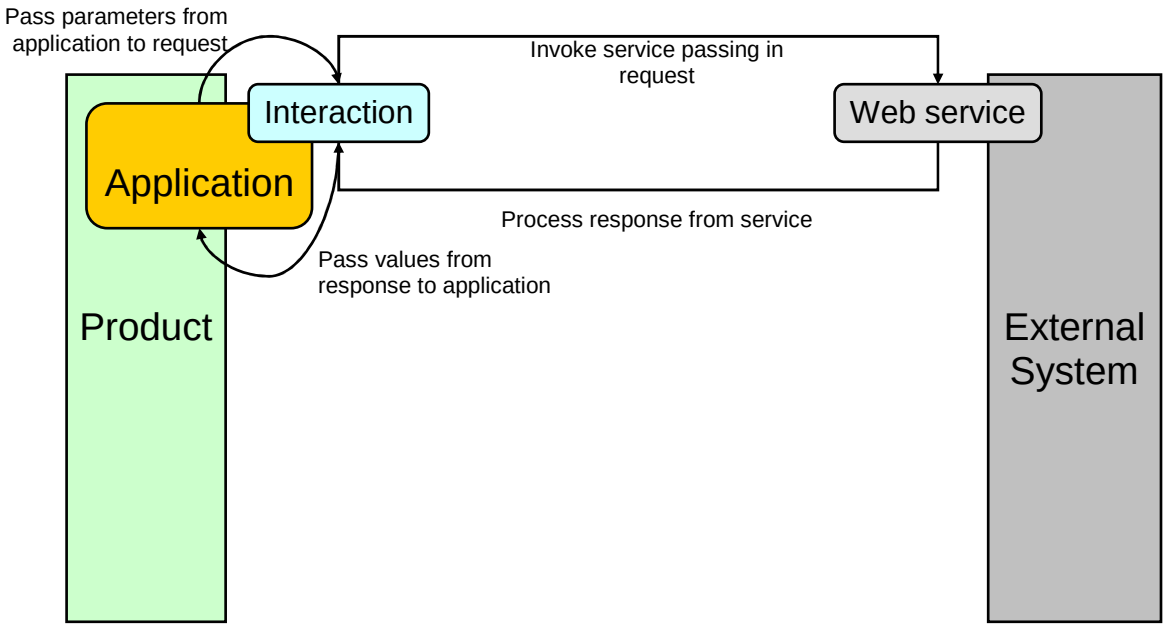


Figure 1 Web service interaction basic functionality

Web service-based interactions must be designed by a practitioner with appropriate skills in web services, XML schemas and Integration Framework technologies. Once the design is complete, the interaction can be made available to end users for execution.

Types of users

Web service interactions target two types of users:

Practitioner, Designer, Consultant, Developer, Business Analyst	This is a resource that has knowledge and experience with web services, XML schemas and Maximo Base Services configuration capabilities. The resource may adopt an iterative approach to creating and testing the appropriate web service interaction.
End users	This is a user that executes web service interactions from a business application for the purpose of performing business tasks. The end user is not aware how the interaction was constructed or what the architecture is supporting execution of such interactions.

Figure 2 illustrates the two types of users.

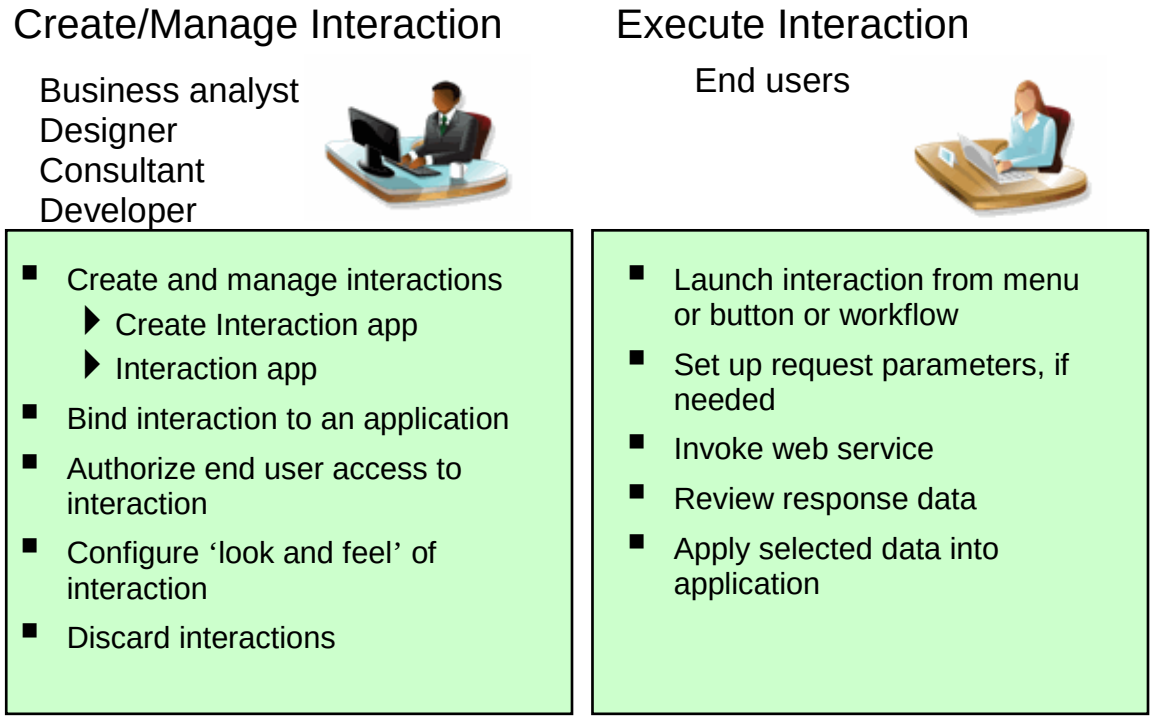


Figure 2 Users exploiting web service interactions

Web service interactions user interface

There are two types of user interfaces supporting web service interactions. One user interface consists of the business applications that enable the creation and management of web service interactions. These are configuration applications exploited by the practitioner. **Figure 3** shows the configuration application user interface to manage existing web service interactions.

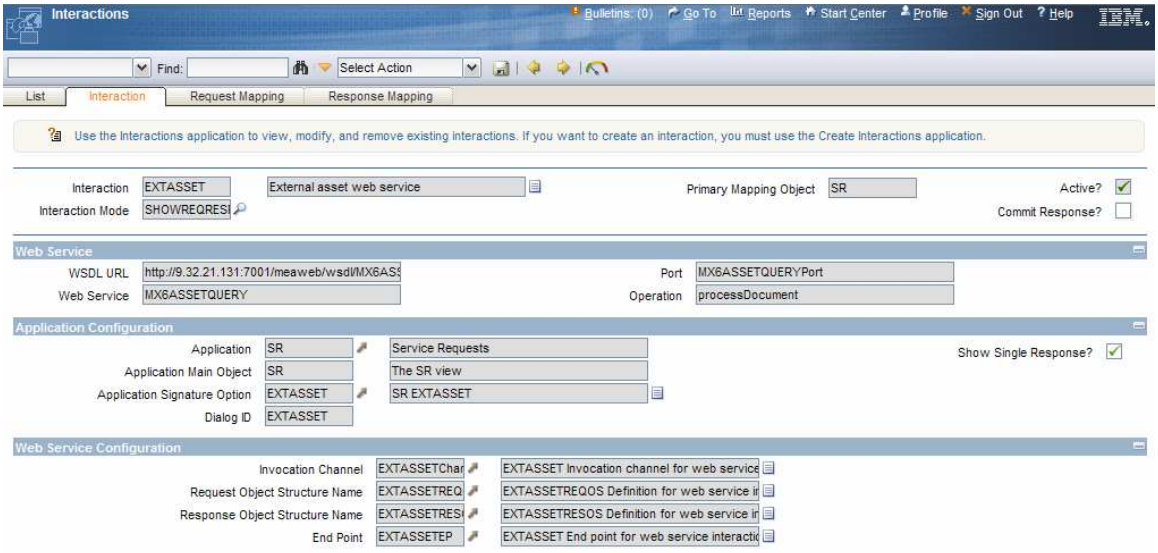


Figure 3 Interactions configuration application

The second user interface is *generated* during the configuration of the web service interaction and resides as part of the business application from which the interaction will be executed. The design of this latter user interface is part of the overall configuration of the web service interaction. The generated user interface is in the form of a popup dialog box. This dialog box can be launched from a Select Action menu item from the launching business application or from a button that is displayed in the launching business application.

Figure 4 shows an example generated interaction user interface.

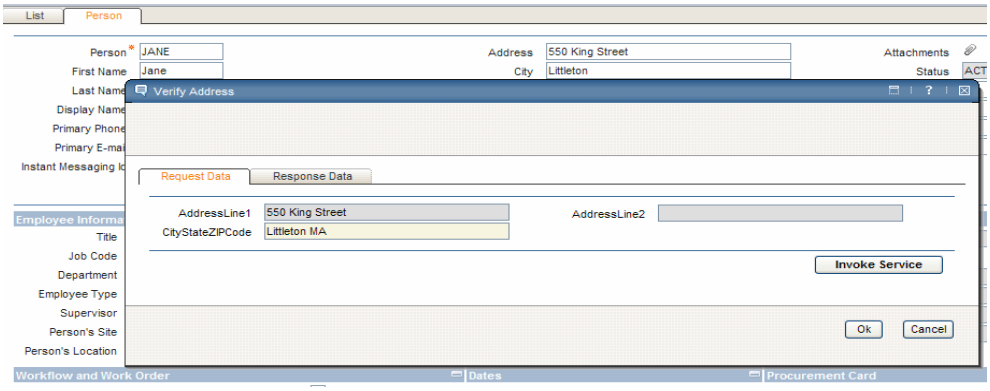


Figure 4 Generated interaction user interface launched from People application

Applications supporting web service-based interactions

Two applications support creation and management of web service-based interactions. They are:

Create Interactions	Application to design and save an interaction
Interactions	Standard application to manage existing interactions

Create Interactions application

This application is accessible from the product's Go To menu. The short cut to this application can be found in the Integration menu under a sub-menu called Interactions. Upon clicking the shortcut, the application is launched and displays the initial screen.

Create Interactions application flow

The application flow of screens is shown in the **Figure 5**.

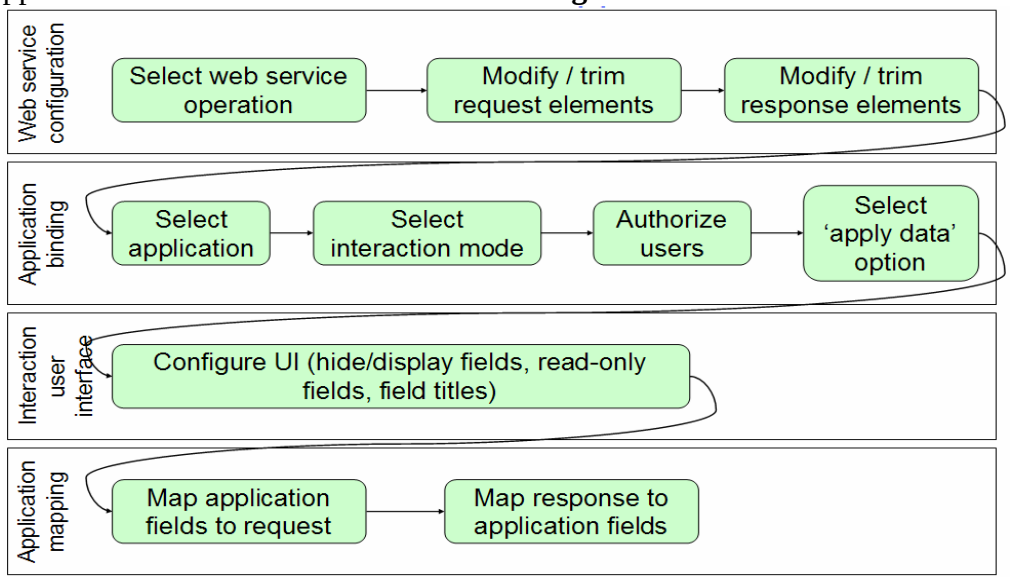


Figure 5 Create Interactions application flow

The initial and subsequent screens are described in the following table:

Screen	Description
Step 1: Select a web service, port and operation for the interaction	This screen is the first step in the initial configuration of the web service interaction. A valid WSDL-based URL must be specified in the WSDL URL field. The URL may represent a http or a file location.
Step 2: Configure the request that will be sent from the application	This step operates on the request elements of the chosen operation. This is an optional step.
Step 3: Configure the response that will be sent back to the application	This step operates on the response elements of the chosen operation. This is an optional step.
Step 4: Select the application	This step configures the application context for the interaction. It determines the specific application from where the interaction will be launched, how the interaction will be launched and which security groups have authorization to launch the interaction from the specified application.
Step 5: Select and define the request fields	This step configures the individual fields that will be displayed in the request user interface generated for the interaction.
Step 6: Select and define the response fields	This step configures the individual fields that will be displayed in the response user interface generated for the interaction.
Step 7: Map information from the application to the web service	This step configures the mapping between application object and the web service request such that data can be passed in from the application to the web service.
Step 8: Map information from the application to the web service	This step configures the mapping between the web service response and the application such that data can be passed back to the application from the web service.
Step 9: Confirm creation of interaction	This screen is the final step in the initial configuration of the web service interaction. The user interface displays various entities that will be constructed if the Create button is clicked.

Step 1 Details – Web service, port and operation

Step 1 in the Create Interactions application is used to specify a web service, select a port from within the web service and select an operation from within the port. Each web service interaction is built on a single operation.

Both SOAP 1.1 and SOAP 1.2 envelopes are supported.

An interaction can be built to support a request-response message pattern. Alternatively, if the web service invocation follows a one-way message pattern, an interaction can be created to support such a web service invocation. If the Process Response check box is checked, the web service does return a response and the interaction will process the response. If the Process Response check box is unchecked, the web service does not return a response or there is no need to process a response.

Step 2 Details – Configure request

Step 2 in the Create Interactions application is used to optionally configure the Request hierarchy. The hierarchy displays the elements of the request and the practitioner can choose to delete elements that are not required for the interaction that is being created. The choice here is based on the integration requirements and the type of data that must be displayed or populated in order to prepare the request. Elements that are marked required in the XML cannot be deleted from the Request hierarchy.

If the Configure button in the Request hierarchy is clicked, the Request Object Structure, Request Objects and Request Attributes sections are refreshed to display the object structure, objects and attributes that will be created in support of the web service request.

It is not mandatory to click the Configure button. Clicking Next in Step 2 will also result in the Request Object Structure, Request Objects and Request Attributes being defined (but not created yet). The practitioner can return to Step 2 from Step 3 to review these definitions.

The request object structure name is based on the interaction name specified in Step 1 followed by the letters ‘REQOS’ representing request Object Structure.

The object names are also based on the interaction name specified in Step 1 followed by a number that makes the object name unique.

Each object in the Request Objects section has a detailed section. Each attribute in the Request Attributes section has a detailed section. The Field Title of the attribute can be edited if required.

The attributes of an object are associated with a data type that is equivalent to the original XML schema data type. **Appendix A** enumerates the XML schema data type and equivalent product data types.

Step 3 Details – Configure response

Step 3 is shown only if Process Response was check box was checked in Step 1. Otherwise, the initial configuration moves onto Step 4.

Step 3 in the Create Interactions application is used to optionally configure the Response hierarchy. The hierarchy displays the elements of the response and the practitioner can choose to delete elements that are not required for the interaction that is being created. The choice here is based on the integration requirements and the type of data that must be displayed or populated in order to prepare the request. Elements that are marked required in the XML cannot be deleted from the Response hierarchy.

If the Configure button in the Response hierarchy is clicked, the Response Object Structure, Response Objects and Response Attributes sections are refreshed to display the object structure, objects and attributes that will be created in support of the web service response.

It is not mandatory to click the Configure button. Clicking Next in Step 3 will also result in the Response Object Structure, Response Objects and Response Attributes being defined (but not created yet). The practitioner can return to Step 3 from Step 4 to review these definitions.

The response object structure name is based on the interaction name specified in Step 1 followed by the letters ‘RESOS’ representing response Object Structure.

The object names are also based on the interaction name specified in Step 1 followed by a number that makes the object name unique.

Each object in the Response Objects section has a detailed section. Each attribute in the Response Attributes section has a detailed section. The Field Title of the attribute can be edited if required.

Step 4 Details – Application selection

Step 4 configures the application launch point for the interaction. This is also termed as the application context. The following table describes the fields that must be populated:

<table>
<tr><th>Field</th><th>Purpose</th></tr>
<tr><td>Application</td><td>This field specifies the application from which the interaction will be launched.</td></tr>
<tr><td>Application Main Object</td><td>This field is automatically populated based on the choice of application. This field is not editable.</td></tr>
<tr><td>Primary Mapping Object</td><td>This field defaults to the same value as Application Main Object. However, the field is editable and a different object can be chosen. The Primary Mapping Object is the object from which values can be passed to the web service request.</td></tr>
<tr><td>Application Signature Option</td><td>This field is automatically populated based on the interaction name specified in Step1. The signature option is required in order to configure the necessary authorizations for end users to launch this interaction from the chosen application. The description for the signature option is also used as the label for the Select Action menu item that is added into the application if the interaction can be launched from the Select Action menu.</td></tr>
<tr><td>Interaction Mode</td><td>This field defaults to a value SHOWREQRESP implying the interaction will display both request and response to end users. If Process Response was not checked in Step 1, the Interaction Mode field will default to SHOWREQONLY. There are four possible values for this field and they are described below:
<table>
<tr><th>Interaction Mode</th><th>Description</th></tr>
<tr><td>SHOWREQONLY</td><td>This value will result in end users seeing only the web service request before the web service is invoked. The response from the web service is not displayed but applied to the application.</td></tr>
<tr><td>SHOWRESPONLY</td><td>This value will result in end users seeing only the web service response after the web service is invoked. The request to the web service is not displayed.</td></tr>
<tr><td>SHOWREQRESP</td><td>This value will result in end users seeing both request and response related to the web service.</td></tr>
<tr><td>SILENT</td><td>This value will result in no user interface for end users. The web service invocation and response are processed behind-the-scenes.</td></tr>
</table>
</td></tr>
<tr><td>Apply Response?</td><td>This field is checked indicating that the response from the web service should be applied back into the application. The data returned from the web service can be potentially inserted or updated back into the application.</td></tr>
</table>

In this step, the Security Groups authorized to execute the interaction are also chosen. By default, the screen displays all of the security groups currently granted access to the chosen application. If required, a subset of these groups can be chosen from the Application Security Groups Authorized to Launch Interaction section.

Step 5 Details – Configure request user interface fields

Step 5 is shown only if the Interaction Mode is either SHOWREQ or SHOWREQRESP – that is, the request is to be displayed to end user. Otherwise, the initial configuration moves onto Step 6.

This step enables the practitioner to select characteristics of individual fields in the request user interface. The request user interface is a tab in the interaction user interface dialog box that enables end users to see the individual fields of the request. For each field, the following can be set:

- Display/Hide the field
- Specify a title for the field
- Set the field read-only or editable

In this step, fields can be configured for each of the objects comprising the request. If there are multiple objects supporting the request, each object must be chosen in order to configure the fields belonging that that object. By default, all the fields belonging to an object are set to display. All the fields are set to be editable in the interaction user interface.

Step 6 Details – Configure response user interface fields

Step 6 is shown only if the Interaction Mode is either SHOWRESP or SHOWREQRESP – that is, the response is to be displayed to end user. Otherwise, the initial configuration moves onto Step 7.

This step enables the practitioner to select characteristics of individual fields in the response user interface. The response user interface is a tab in the interaction user interface dialog box that enables end users to see the individual fields of the response. For each field, the following can be set:

- Display/Hide the field
- Specify a title for the field
- Set the field read-only or editable

In this step, fields can be configured for each of the objects comprising the response. If there are multiple objects supporting the response, each object must be chosen in order to configure the fields belonging that that object. By default, all the fields belonging to an object are set to display. All the fields are set to be editable in the interaction user interface.

Step 7 Details – Mapping application objects to request

This step enables the practitioner to map objects and attributes of the business application to the objects and attributes that represent the web service request. This mapping is driven by the Primary Mapping Object that was chosen in Step 4. The primary mapping object serves as the root for the mapping.

The two key sections in this step are the Request Objects section and the Request Attribute Mapping section. For mapping, a request object is chosen from the Request Objects section. In order to set up a mapping at the object level, the mapping user interface should be interpreted as follows: "The Request Object <x> should be mapped or set from the Application Object <y>".

Figure 6 shows the Request Objects and Request Attribute Mapping sections for an example web service interaction configuration:

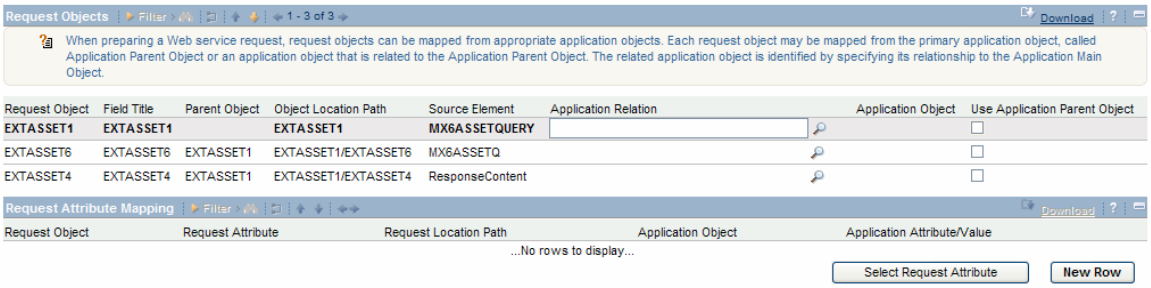


Figure 6 Request Objects and Request Attribute Mapping

If the application object is the same as the Primary Mapping Object specified earlier, then the Use Application Parent Object check box can be checked.

Figure 7 shows the Request Objects section once Use Application Parent Object has been checked:

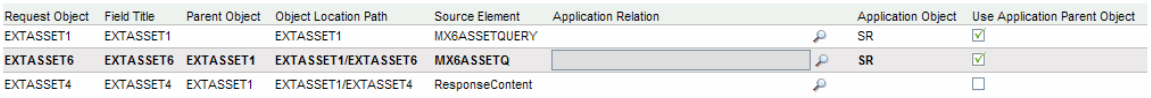


Figure 7 Request Objects specified

If the application object is different from the Primary Mapping Object specified earlier or any parent object, then the relationship linking the parent object and application object should be specified in the Application Relation field. A lookup is provided in order to facilitate selecting the right relationship.

In order to map attributes, first select the appropriate request object from the Request Objects section. Click the New Row or Select Request Attribute buttons in order to select an attribute. The mapping user interface for attributes should be interpreted as “The Request Attribute <xx> should be mapped or set from the Application Attribute <yy>”.

When the New Row button is clicked, a new row is displayed in the Request Attribute Mapping section and the Request Attribute and Application Attribute/Value fields are editable. The associated lookups can be utilized to select the appropriate attribute of the request and the application object. In addition, the Application Attribute/Value field can hold a literal value enclosed in single quotes. The standard dot (‘.’) notation may also be used to map an attribute from a related object.

Figure 8 shows the Request Attributes Mapping section when a new row has been inserted:

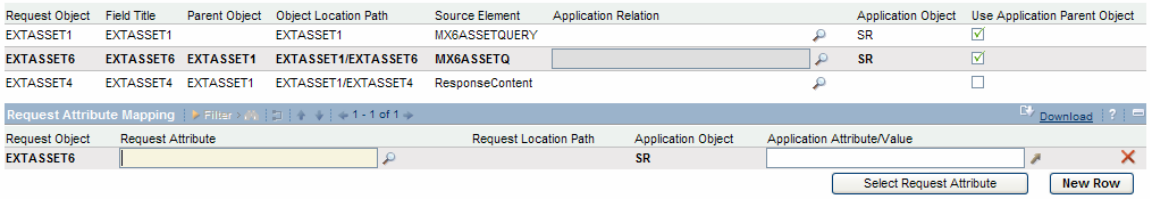


Figure 8 Request Attribute Mappings

Figure 9 shows the Request Attribute Mapping section after the new row has been populated:

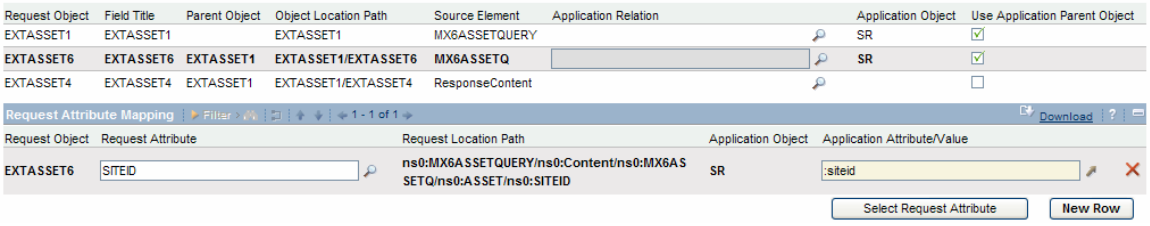


Figure 9 Request Attribute Mappings populated

The attribute mapping example shown should be interpreted as “The Web Service Request Attribute SITEID is mapped from the Application object SR and its attribute `siteid`.”

Attribute Mappings can be deleted by clicking the Mark Row for Delete icon to the right of each row in the Request Attribute Mapping section.

It is not mandatory to map every web service request element from the application. Mappings are set up where appropriate. Mappings make it easier for end users to use the web service interaction because specified application field values are automatically brought into the web service request when the interaction is executed.

A Java class can be authored if the mapping constructs in Step 7 are insufficient to perform the type of mapping desired. The class should be authored using appropriate Java-development tools and the class file built into the product Enterprise Archive (EAR). Standard deployment procedures should be followed to deploy this EAR file such that the class file is available for the Integration Framework to execute. The fully qualified name of the Java class should be specified in the Mapping Custom Class field.

Mapping can be implemented using a combination of Java class and configured mappings.

Mappings can be added or modified using the Interactions application even after the interaction has been created.

Step 8 Details – Mapping response to application objects

This step enables the practitioner to map objects and attributes of the web service response to the objects and attributes of the business application. This mapping is driven by the Primary Mapping Object that was chosen in Step 4. The primary mapping object serves as the root for the mapping.

The two key sections in this step are the Response Objects section and the Response Attribute Mapping section. For mapping, a response object is chosen from the Response Objects section. In order to set up a mapping at the object level, the mapping user interface should be interpreted as follows: “The Application Object <a> should be mapped or set from the Response Object <b>”.

Figure 10 shows the Response Objects and Response Attribute Mapping sections for an example web service interaction configuration:

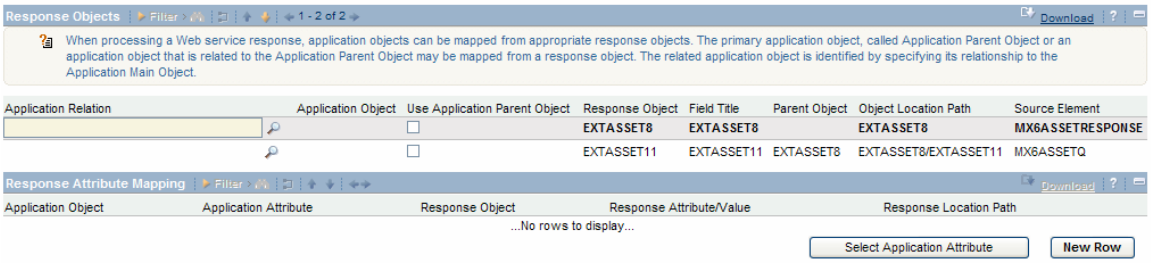


Figure 10 Response Objects and Response Attribute Mapping

If the application object is the same as the Primary Mapping Object specified earlier, then the Use Application Parent Object check box can be checked.

Figure 11 shows the Request Objects section once Use Application Parent Object has been checked:

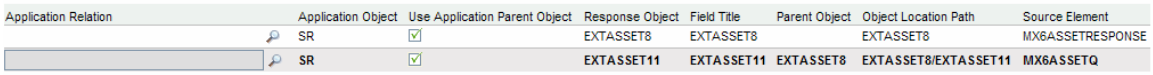


Figure 11 Response Objects specified

If the application object is different from the Primary Mapping Object specified earlier or any parent object, then the relationship linking the parent object and application object should be specified in the Application Relation field. A lookup is provided in order to facilitate selecting the right relationship.

In order to map attributes, first select the appropriate response object from the Response Objects section. Click the New Row or Select Request Attribute buttons in order to select an attribute. The mapping user interface for attributes should be interpreted as "The Application Attribute <aa> should be mapped or set from the Response Attribute <bb>".

When the New Row button is clicked, a new row is displayed in the Response Attribute Mapping section and the Application Attribute and Response Attribute/Value fields are editable. The associated lookups can be utilized to select the appropriate attribute of the request and the application object. In addition, the Response Attribute/Value field can hold a literal value enclosed in single quotes.

Figure 12 shows the Response Attributes Mapping section when a new row has been inserted:

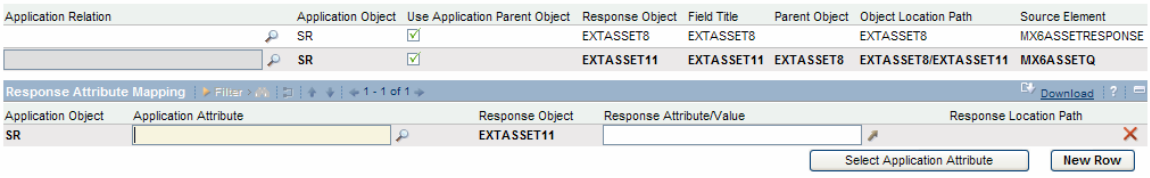


Figure 12 Response Attributes Mapping

Figure 13 shows the Response Attribute Mapping section after the new row has been populated:

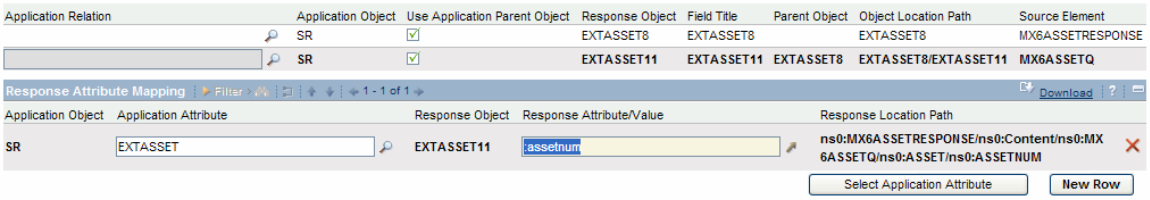


Figure 13 Response Attributes Mapping populated

The attribute mapping example shown should be interpreted as "The Application Attribute EXTASSET is mapped from the Response Object EXTASSET11 and its attribute `assetnum`."

Attribute Mappings can be deleted by clicking the Mark Row for Delete icon to the right of each row in the Response Attribute Mapping section.

Step 9 Details – Confirmation and creation of interaction

This is the final review step in the creation of the interaction. All fields displayed in this step are for review purposes only. With this step, the practitioner can either cancel the creation by clicking the Cancel button or proceed by clicking the Create button.

Clicking Create button will construct various artifacts that serve as the building blocks of the interaction. These artifacts include:

Generated Artifact	Purpose
End Point	Integration artifact that will perform the actual web service invocation. The name of the end point is the same as the name of the interaction followed by the letters 'EP' standing for end point.
Invocation Channel	Integration artifact that will perform the request and response processing in the context of the web service invocation. The name of the invocation channel is the same as the name of the interaction followed by the word 'Channel'.
Request Object Structure	Integration artifact that will prepare the request for the web service. The name of the request object structure is the same as the name of the interaction followed by the letters 'REQOS' standing for request object structure.
Response Object Structure	Integration artifact that will receive the response for the web service. The name of the response object structure is the same as the name of the interaction followed by the letters 'RESOS' standing for request object structure.
Request Objects	Non-persistent business objects that will hold the request information in the context of web service invocation. Mappings from the application objects will bring application values into the request objects. The names of the request objects are the same as the name of the interaction followed by a numeric suffix.
Response Objects	Non-persistent business objects that will hold the response information in the context of web service invocation. Mappings to the application objects will bring response values into the application objects. The names of the response objects are the same as the name of the interaction followed by a numeric suffix.

Generated Artifact	Purpose
Relationships	Request and response objects will utilize relationships to determine parent and child. Such relationships will be generated. The names of the relationships will be the same as the child object in the relationship.
Domains	If the XML schema contains enumerations for an element that is part of the web service request, an ALN or numeric domain will be generated to hold the values in the enumeration depending upon the data type of the attribute it is associated with. The name of the domain will include the name of the attribute it is associated with. The generated domains are not displayed on Step 9.
Application Signature Option	Security artifact that can be configured to grant security groups authorization to execute the interaction. This signature option can also be configured to associate user interface such as a button if the interaction is to be launched from a button in the application. The name of the signature option is the same as the name of the interaction.
Menu Option	Menu entry that will be used to launch the interaction from the Select Action menu of the business application. The name of the menu option is the same as the name of the interaction.
Request/Response Dialog	The user interface that end users will interact with when performing the web service invocation. The dialog resides as part of the presentation for the application from where the interaction is launched. The dialog can be viewed and/or modified from the Application Designer. The name of the dialog is the same as the name of the interaction.

User interface behavior of the Create Interaction application

At any time during the initial configuration of the interaction, a practitioner may click Cancel to exit the application. All configuration done up until that point will be lost. When the Cancel button is clicked, the practitioner is returned to the Start Center.

Once the interaction has been successfully created, the application displays a confirmation message “Interaction <ccc> has been created. You can return to the Start Center or you can go to the Interactions application to make changes to this interaction.” The message box offers practitioners the choice of returning to Start Center or launching the Interactions application.

Interactions application

The Interactions application provides access to and management of existing interactions. There are four tabs in this application as listed below:

Tab	Purpose
List	Enumerates the interactions that have been already created; specifies if the interaction is active or inactive
Interaction	Provides the details of the web service this interaction will invoke; also enumerates all of the artifacts that were generate in support of this interaction
Request Mapping	Enables the management request mappings – mappings can be created, modified or deleted. The user interface is very similar to Step 7 of the Create Interactions application.
Response Mapping	Enables the management response mappings – mappings can be created, modified or deleted. The user interface is very similar to Step 8 of the Create Interactions application.

Additional Functions in the Interactions application

Interaction main tab

Most of the fields displayed on the main tab are read-only. However two fields can be configured:

Interaction tab field	Purpose
Commit Response?	Checking this field implies that any data applied into the application based on the web service response will be committed as soon as they are applied. If this field is unchecked, data applied into the application must be manually saved by the end user who launched the interaction. By default, this field is unchecked.
Show Single Response?	Checking this field implies that a single result record returned by the web service will be displayed in the interaction user interface. If this field is unchecked, single result record returned by the web service will be automatically applied into the application where the interaction was launched from. By default, this field is checked.

Global Schema Policies

An XML-based schema is inherent to a web service. The schema describes data types, elements and other characteristics of various components of the web service. A powerful schema language enables the creation of complex schemas.

Durable web service interactions depend upon the proper resolution and interpretation of various schema constructs. A fixed set of schema policies are available in order to ensure proper resolution and interpretation. They are accessible from the Global Schema Policies Select Action menu item in Interactions application. A popup dialog box enumerates the policies that are required when creating an interaction. Each schema policy is associated with one or more policy parameters. Policy parameters have default values that can be changed, if required. The following table lists each policy and its purpose:

Policy	Purpose	Policy Parameter	Default Value
Any Element Policy	Processing of 'any' elements in the XML schema underlying a web service; the default value of 1 implies that 'any' elements in the schema will be skipped for the purpose of creating an interaction	EXCLUDE_ANY_ELEMENT	1
Attribute Exclusion Policy	Determines if attributes of elements should be excluded from schema processing; the default value of 1 implies attributes of elements will not be processed	EXCLUDE_ATTRIBUTES	1
Default Attribute Length	Determines the length of an attribute that is created as part of an object to support the interaction	DEFAULT_LENGTH	256
Object Recurrence Policy	If set to 1 the recurring type definition will be ignored and only the first occurrence of the type will be utilized for configuring the interaction If set to 0 will report an error and stop processing the schema if a recursive type definition is detected	SKIP_RECURRENCE	1
Simple Type Policy	If set to 1 simple list type will be processed as simple atomic type. This means there can be only one value entered for that list type element. The type of the element is retained. If set to 0 the simple list type is treated as a String type	TREAT_SIMPLE_LIST_TYPE_AS_ATOMIC	0

If any of the out of the box default values were modified, the original defaults can be applied by selecting the appropriate policy and clicking the Restore Defaults button.

Activate/Deactivate Interaction
This action can be used to determine whether an interaction is available for execution by end users. If an interaction has been deactivated, end users attempting to launch the interaction from the parent application will receive an error message indicating the interaction has been deactivated. This function can be exploited when there is a need to create, modify or delete mappings after the interaction was created. The changes to mappings can be performed with the interaction deactivated. This prevents a web service invocation from occurring during modifications that may result in invocation or processing errors.

Validate Interaction
This action can be used to verify the integrity of the interaction. The validate function iterates through all the generated artifacts that are required for the interaction to execute correctly and verifies they exist. The function lists all the artifacts that are required for the interaction including those that were found and those that were not found. The interaction cannot be activated if the validation fails.

Delete Interaction
This action can be used to delete an existing interaction. All generated artifacts are deleted with the exception of domains.

Logging configuration time and execution time activities

The initial configuration of an interaction can be complex depending upon the complexity of the web service to be invoked. It is useful to log the configuration activity and maintain a log file of what was configured. A dedicated interaction logger is available for this purpose. The name of the logger is ‘interaction’ and the Logging application can be configured to set the ‘interaction’ logger to DEBUG level in order to generate the detailed log of configuration.

Once an interaction has been created, the execution of the interaction can be also be logged using existing Integration Framework loggers. The Logging application can be configured to set the ‘integration’ logger to INFO or DEBUG level in order to generate run-time logs covering execution of the interaction.

Limitations

The following are known limitations of web service-based interactions:

1. Only WSDL 1.0 standard is supported.
2. If the currently logged in user has been granted authorization to execute the interaction, then the current user must log out and log back in for his/her authorization to take effect.
3. The interactions and generated artifacts are all based on the base language of the product environment. Users who log in to the product with non-base language will see the base language user interface of the interaction. The user interface for the interaction should be localized if required using IBM-provided localization tools.
4. Interactions can be created only in the base language of the product environment. A user logged in to the product with a different language than base language will not be able to create an interaction. An error message will be displayed in the Create Interactions application.
5. There are several XML schema limitations:
    a. Schema 'any' element should be resolved prior to creating the interaction
    b. Recursive schema elements are not supported
    c. Substitution groups are not supported
    d. Sequencing is not supported
6. If any of the generated artifacts are deleted from the product using the corresponding configuration application (for example, the interaction dialog is deleted from the Application Designer application), then the interaction will not function as it is missing a required component.

Appendix A. Schema Types and equivalent Maximo Data Types

Schema Types	Maximo Data Type
Boolean	YORN
Byte, Base64Bytes, HexBytes	BLOB
Datetime	DATETIME
Date	DATE
Time	TIME
Double, Float	FLOAT
Short, Long, Integer	INTEGER
BigInteger	
If no length specified	ALN
If length specified	INTEGER
BigDecimal	
If no length specified	ALN
If length specified	DECIMAL
String and other data types	ALN

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks