

blueprism[®]

Robotic Process Automation Software

Web Services

USER GUIDE

Major Version: 6

Document Revision: 1.0

For more information please contact:

info@blueprism.com | UK: +44 (0) 870 879 3000 | US: +1 888 757 7476

www.blueprism.com

Contents

1.	Introduction	3
1.1.	Audience	3
2.	Executive Summary	4
3.	Exposing Web Services	5
3.1.	Providing Access to the Web Services	6
3.2.	Defining Web Service Parameters	7
3.3.	Web Service Encoding Types	7
3.4.	Walkthrough: Expose a Business Object as a Web Service	8
3.5.	Walkthrough: Expose a Process as a Web Service	11
4.	Consuming Third-Party Web Services	14
4.1.	Accessing Third-Party Web Services	15
4.2.	Web Service Parameters	16
4.3.	Walkthrough: Setup a Third-Party Web Service in Blue Prism	18
4.4.	Walkthrough: Use a Third-Party Web Service in Blue Prism	22
5.	Advanced Topics.....	26
5.1.	Design Considerations for a Blue Prism Web Service Interface	26
5.2.	Securing Exposed Blue Prism Web Services.....	34
5.3.	Consume Web Services using Code Stage Based Business Objects	35
5.4.	Consume an Exposed Blue Prism Web Service using Visual Studio	45
6.	Frequently Asked Questions	54
7.	Support.....	55

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third party without the written consent of an authorised Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited.

© Blue Prism Limited, 2001 – 2016

®Blue Prism is a registered trademark of Blue Prism Limited

All trademarks are hereby acknowledged and are used to the benefit of their respective owners.
Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, Centrix House, Crow Lane East, Newton-le-Willows, WA12 9UY, United Kingdom
Registered in England: Reg. No. 4260035. Tel: +44 870 879 3000. Web: www.blueprism.com

1. Introduction

이 문서는 웹 서비스 환경에서 사용될 때 Blue Prism 의 기능에 대한 개요를 제공합니다.

이 가이드는 다음과 같은 정보를 제공합니다.

- Blue Prism Business Object 를 웹 서비스로 노출합니다.
- Blue Prism Process 를 웹 서비스로 노출합니다.
- Blue Prism Business Object 또는 프로세스에서 타사의 웹 서비스를 사용합니다.

다음과 같은 고급 주제도 포함됩니다.

- Blue Prism 웹 서비스 인터페이스에 대한 디자인 고려 사항
- 프록시를 사용하여 Blue Prism 웹 서비스를 노출합니다
- 다형성 구조를 사용하여 데이터를 제공하는 타사의 웹 서비스와 상호 작용하는 방법.
- 노출된 Blue Prism 웹 서비스와 상호 작용할 수 있는 독립 웹 서비스 클라이언트를 Visual Studio 에서 만드는 방법.

1.1. Audience

이 문서는 솔루션 설계자 및 디자이너, 개발자 및 Blue Prism 공인 개발자에게 정보를 제공하기 위한 것입니다.

2. Executive Summary

Blue Prism 은 GUI (Graphical User Interface)를 통해 Windows PC 에서 액세스할 수 있는 모든 응용 프로그램을 자동화하도록 설계되었지만 사용자가 웹 서비스를 통해 자동화하고 통합할 수 있는 여러가지 시나리오가 있습니다.

Blue Prism 은 웹 서비스 연결을 지원함으로써 다음과 같은 일도 할 수 있습니다.

- 타사 응용 프로그램 또는 개발자가 다음과 같은 목적으로 Blue Prism 내에서 Business Objects 및 프로세스를 활용하고 기동할 수 있습니다.
- Blue Prism 에 의해 이미 자동화된 레거시 시스템과 상호 작용합니다.
- 미리 결정된 일정에 따라 처리할 준비가 된 적절한 대기열에 작업 항목을 추가합니다.
- Blue Prism 에서 즉시 처리할 작업 항목을 트리거합니다(리소스 가용성에 따라 다름).
- Blue Prism 이 자동화한 시스템에서 데이터를 검색합니다.
- Blue Prism 에서 정보를 검색합니다(예: 작업 대기열, 일정, 작업 내역 등의 세부 정보).
- Blue Prism Business Objects 및 프로세스는 게시된 웹 서비스를 사용하여 타사 시스템과 상호 작용합니다.

웹 서비스 통합은 전통적인 소프트웨어 개발 기술이므로 그 사용은 다음과 같은 높은 수준의 고려 사항에 따라 평가되어야 합니다.

- 수요가 많은 기간 동안에 발생하는 주요 시스템에 대한 성능 영향.
- 프레젠테이션 계층에서 구현된 모든 데이터 유효성 검사가 웹 서비스 상호 작용에 수동으로 적용되도록 합니다(예:이 유효성 검사 논리가 웹 서비스 사용을 통해 우회되지 않는지 확인).
- IT 거버넌스 수준은 GUI 기반 자동화에 필요한 수준과 다를 수 있습니다. 종종 웹 서비스는 애플리케이션의 그래픽 인터페이스를 통해 작업하는 최종 사용자가 얻을 수 있는 것보다 더 광범위한 기능을 제공합니다.

3. Exposing Web Services

일단 구현되면, Blue Prism 은 Blue Prism 에서 제공하는 가상 인력이 처리하고 작업할 수 있는 다양한 시스템 및 비즈니스 프로세스에 대한 자세한 정보를 포함하는 안전한 저장소가 됩니다.

이 정보에는 Blue Prism 런타임 리소스가 프로세스를 성공적으로 완료하기 위해 따라야 하는 규칙, 결정 및 절차에 대한 세부 사항뿐만 아니라, 다양한 시스템 및 기술에서 수행할 수 있는 모든 사용 가능한 작업이 포함됩니다.

Blue Prism Business Objects 및 프로세스를 노출함으로써, 타사 시스템 및 개발자는 이러한 웹 서비스를 호출하여 다양한 목적으로 Blue Prism 의 기능과 유연성을 활용할 수 있습니다.

- Blue Prism 에 의해 이미 자동화된 레거시 시스템과 상호 작용합니다.
- 미리 결정된 일정에 따라 처리할 준비가 된 적절한 대기열에 작업 항목을 추가합니다.
- Blue Prism 에서 즉시 처리할 작업 항목을 트리거합니다(리소스 가용성에 따라 다름).
- Blue Prism 이 자동화하는 시스템에서 데이터를 검색합니다.
- Blue Prism 에서 정보를 검색합니다(예: 작업 대기열, 일정, 작업 내역 등의 세부 정보).

타사 시스템 또는 개발자가 이러한 방식으로 Blue Prism 과 상호 작용할 수 있도록 하는 접근 방식은 다음과 같은 여러 이점을 제공합니다.

- 모든 타사 시스템과 개발자는 Blue Prism 에 설계된 규칙을 적용하는 방법을 통해 주어진 시스템에 연결합니다.
- 시스템이 자동화되는 기존 방식으로 공통성을 적용합니다(예: 각 개발자 또는 시스템이 따르는 것과 다른 규칙 및 경로 제거).
- Blue Prism 은 자동화되는 기술(예: 메인프레임, 웹 애플리케이션, .NET 애플리케이션, 엑셀, SQL 데이터베이스 등)과 관계없이 자동화되는 모든 시스템에 공통 인터페이스를 제공합니다.
- 인터페이스에 필요한 변경 사항을 적용하는 공통 방법을 제공하고, 기존 내부 변경 관리 정책에 맞게 조정합니다. 또한 필요한 작업 및 프로세스에 대한 모든 변경 사항은 해당 팀이 시각적 디자인너를 사용하여 중앙에서 관리하고 제어합니다.

Blue Prism 웹 서비스는 기본적으로 SOAP, WSDL 기반 서비스입니다. RESTful Blue Prism 웹 서비스 생성에 대한 기본 지원은 제공되지 않습니다.

3.1. Providing Access to the Web Services

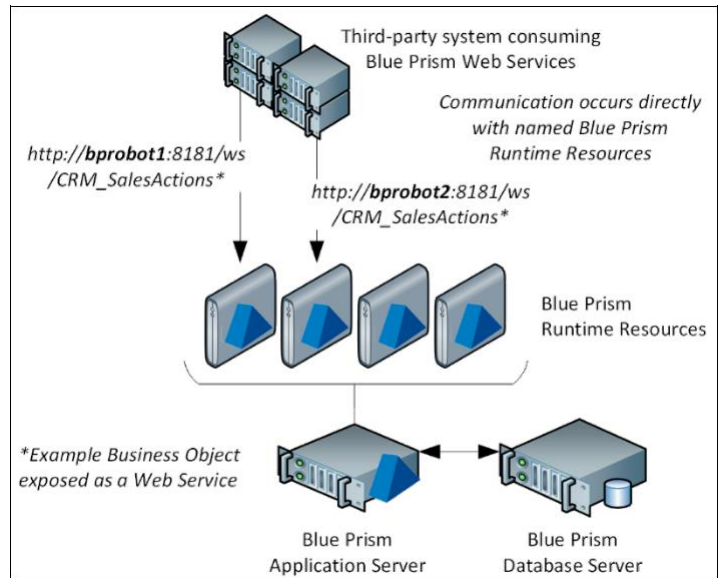
Blue Prism 웹 서비스에 대한 액세스를 허용하려면 다음 항목을 검토해야 합니다.

- **Network connectivity to the Blue Prism runtime resources**

웹 서비스와 상호 작용할 시스템은 지정된 Blue Prism 리소스에 대한 네트워크 연결이 필요합니다.

기본적으로 Blue Prism 웹 서비스는 TCP 트래픽을 위해 포트 8181 에서 사용할 수 있지만 재구성이 가능합니다.

인증서 기반 암호화를 사용하여 이러한 연결을 보호하는 방법에 대한 자세한 내용은 **Securing Exposed Blue Prism Web Services** 을 참조하십시오.



- **Security permissions to interact with the web services**

시스템이 Blue Prism 웹 서비스를 호출할 때, Blue Prism 액세스 자격 증명을 제공하여 시스템 또는 사용자에게 작업을 수행할 수 있는 적절한 권한을 제공해야 합니다.

웹 서비스와 상호 작용할 각 타사 공급자에 대해 사용자 계정을 설정하는 것이 좋습니다. 이 사용자 계정에는 Blue Prism Business Objects 및 프로세스를 성공적으로 실행하는 데 필요한 최소 보안 권한이 할당되어야 합니다. (예: 프로세스에서 제한된 자격 증명 정보에 액세스해야 하는 경우, 타사 시스템에서 사용하는 계정에 이 자격 증명을 활용하려면 적절한 권한이 필요합니다).

Blue Prism 이 Single Sign-on 을 위해 Active Directory 와 통합된 경우 연결 시스템은 적절한 Blue Prism 권한이 있는 도메인 계정에 대한 자격 증명을 제공해야 합니다.

- **Web service definition (WSDL)**

타사 시스템 또는 개발자는 일반적으로 적절한 웹 서비스를 위해 WSDL 에 액세스해야 합니다. 런타임 리소스의 컴퓨터 이름과 포트가 식별되면 다음 URL 이 사용 가능한 각 웹 서비스의 이름, 간단한 설명 및 WSDL 의 주소를 제공합니다. WSDL 에는 웹 서비스의 경로가 포함됩니다.

http://[machine name]:[port]/ws/

Blue Prism 런타임 리소스가 인증서 기반 암호화로 구성된 경우 접두사는 https 가 됩니다.

3.2. Defining Web Service Parameters

Blue Prism 웹 서비스는 입력 매개 변수를 받아들이고 출력 매개 변수를 반환하도록 구성할 수 있습니다. Blue Prism 웹 서비스의 일부로 구성될 매개 변수는 각 비즈니스 객체 및 프로세스에 정의된 입력 및 출력 매개 변수를 기반으로 합니다.

표시된 Blue Prism 데이터 유형은 입력 및 출력 매개 변수 모두에 대해 지원되며 적절한 웹 서비스 XSD 데이터 유형에 매핑됩니다.

Blue Prism 은 입력 및 출력 매개 변수로 데이터 컬렉션(목록) 사용을 지원합니다. 단, 수집 구조가 미리 결정되고 지원되는 데이터 유형의 데이터 항목으로만 구성됩니다.

Blue Prism Data Type	Web Service XSD Data Type
date	date
dateTime	dateTime
flag	boolean
number	decimal
text	String
time	time
timespan	duration
binary	base64binary

3.3. Web Service Encoding Types

지정된 웹 서비스를 표시하는 데 사용되는 인코딩 유형은 정의된 입력 및 출력 매개 변수의 특성을 기반으로 합니다.

기본적으로 Blue Prism 웹 서비스는 **RPC/Encoded** 웹 서비스로 노출되지만, 웹 서비스의 입력 또는 출력 매개 변수가 Blue Prism 데이터 수집을 활용하도록 구성된 경우, 해당 작업은 **Document/Literal** 인코딩 오퍼레이션으로 노출됩니다.

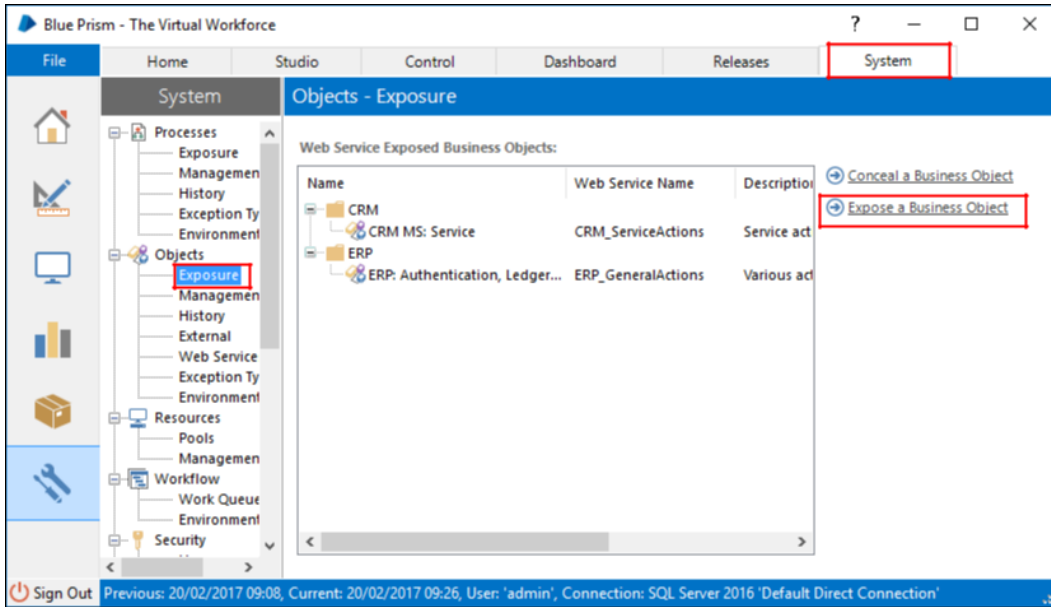
Blue Prism Business Objects 에는 **RPC/Encoded** 오퍼레이션으로 인코딩되는 여러 기본 작업이 자동으로 포함됩니다. 즉, 노출된 Business Object 의 WSDL 에는 항상 하나 이상의 **RPC/Encoded** 오퍼레이션이 포함됩니다.

노출된 Blue Prism 웹 서비스는 **UTF-8** 문자 인코딩으로 제공됩니다.

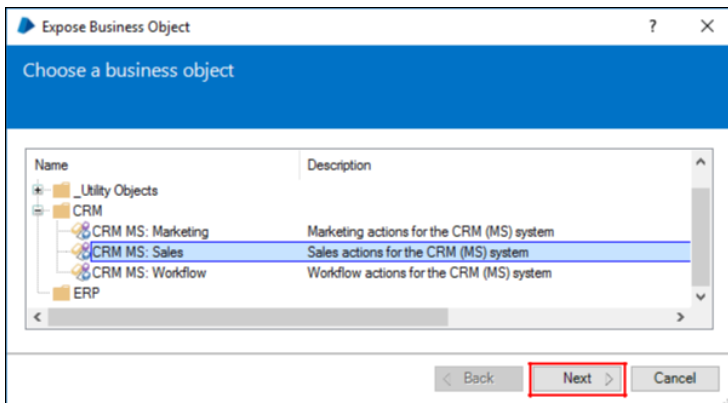
3.4. Walkthrough: Expose a Business Object as a Web Service

이 연습에서는 Blue Prism Business Object 를 웹 서비스로 사용할 수 있도록 만드는 데 필요한 단계를 제공합니다. Business Object 가 웹 서비스로 사용 가능한지 확인하고 WSDL 에 액세스하는 방법도 제공됩니다.

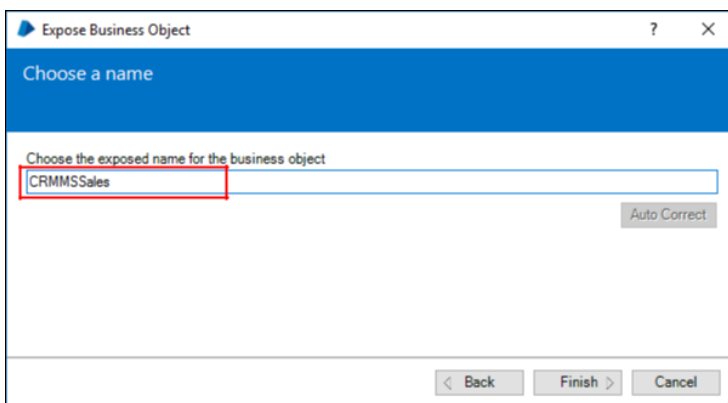
1. System Manager 에 들어가서 Objects 영역을 선택한 다음 Expose a Business Object 를 클릭합니다.



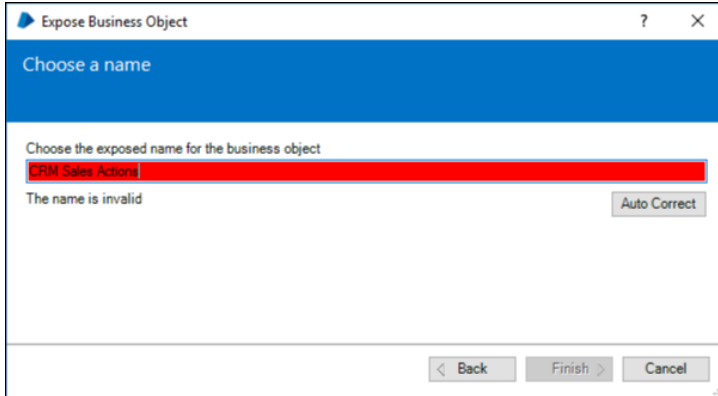
2. 현재 웹 서비스로 노출되지 않은 비즈니스 오브젝트 목록이 표시됩니다. 노출할 비즈니스 오브젝트를 선택하고 Next 를 클릭하십시오.



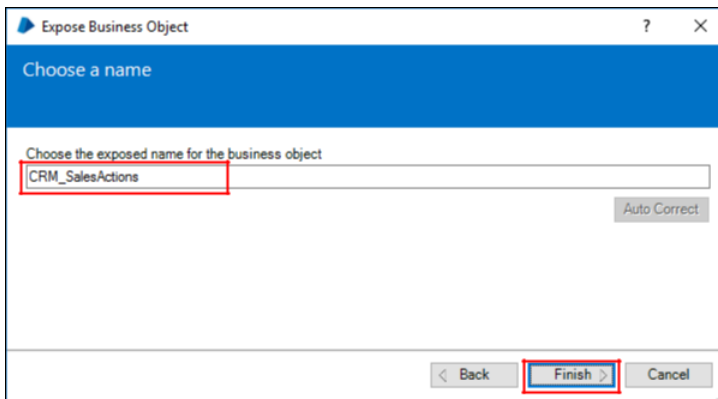
3. 마법사는 Business Object 의 이름을 사용하여 웹 서비스에 사용할 적절한 이름을 제안합니다.



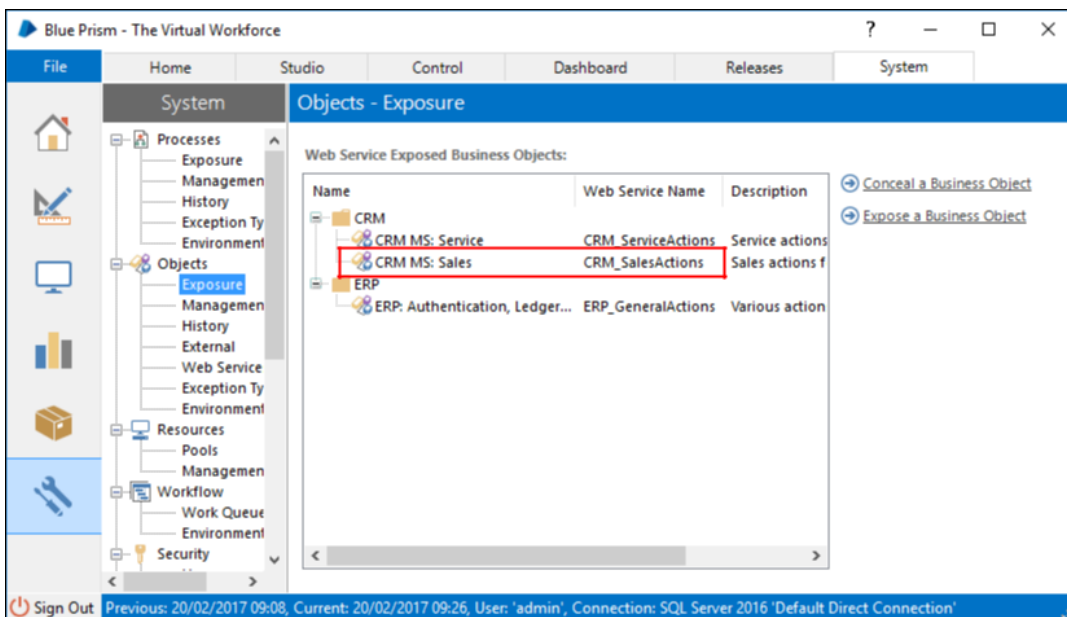
이 시점에서 이름을 변경할 수 있지만 이름이 이전에 노출된 Blue Prism Web Service 의 중복이거나 유효하지 않은 문자가 감지되면 경고가 표시됩니다. 자동 수정을 클릭하면 잘못된 문자가 제거되고 이름이 고유한지 확인합니다.



- 이름이 허용되면 Finish 를 클릭합니다.



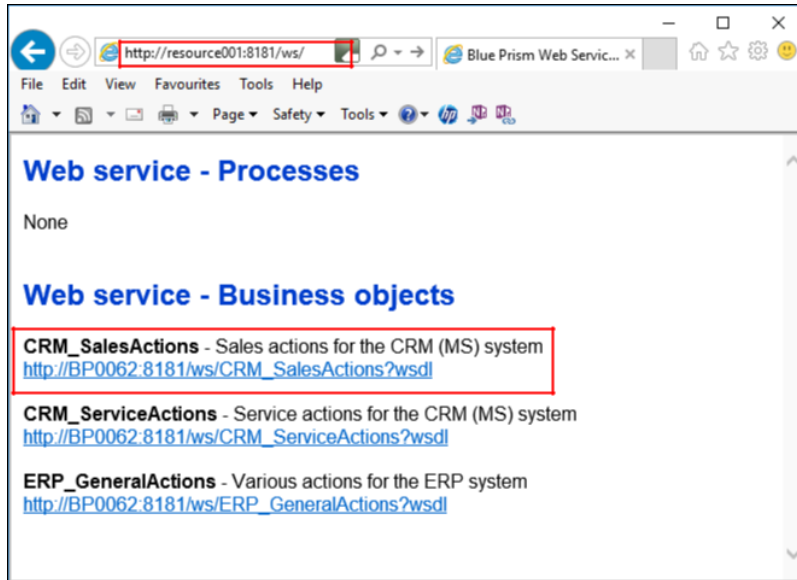
- 새로 노출된 웹 서비스 이름이 이제 Business Object 의 이름과 함께 목록에 나타납니다.



6. Blue Prism 에서 노출되는 웹 서비스는 WSDL 의 주소를 제공하는 다음 URL 을 사용하여 확인할 수 있습니다.
http://[machinename]:<port>/ws/.

이 예에서 컴퓨터 이름은 resource001 이고 기본 포트인 8181 이 사용 중입니다.

http://resource001:8181/ws/

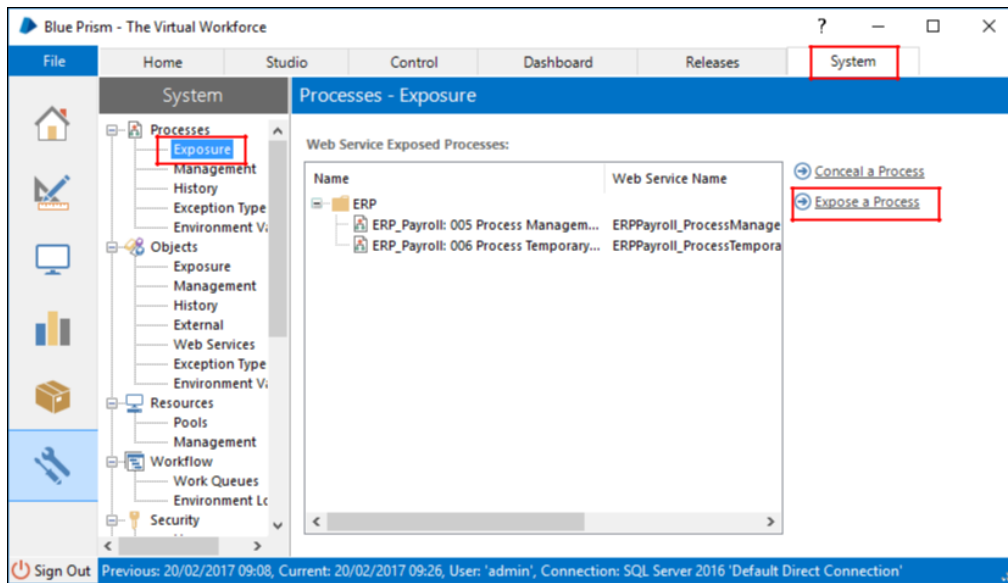


Blue Prism 런타임 리소스가 인증서 기반 암호화로 구성된 경우 접두사는 **https** 가 됩니다.

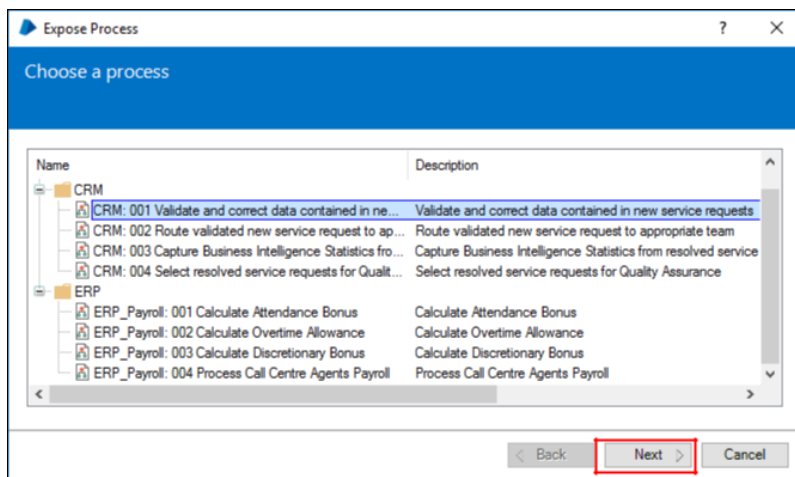
3.5. Walkthrough: Expose a Process as a Web Service

이 연습에서는 Blue Prism 프로세스를 웹 서비스로 사용할 수 있도록 만드는 데 필요한 단계를 제공합니다. WSDL 에 액세스하는 방법을 웹 서비스로 사용할 수 있는지 확인하는 단계도 제공됩니다.

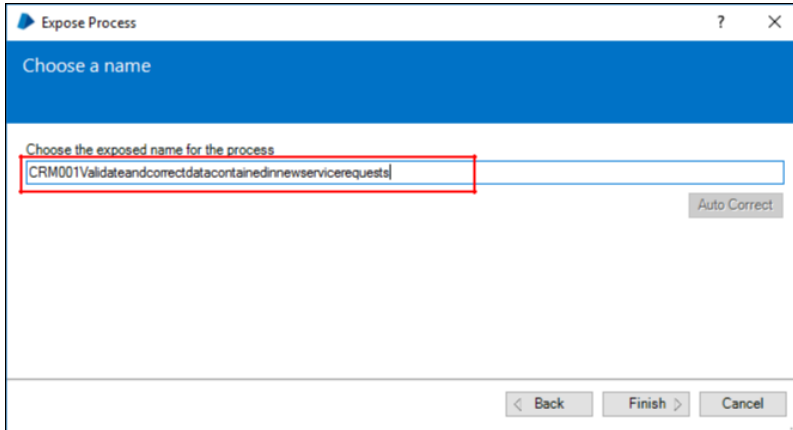
1. System Manager 에 들어가서 Processes 영역을 선택하십시오. Exposure and Expose a Process 를 클릭합니다.



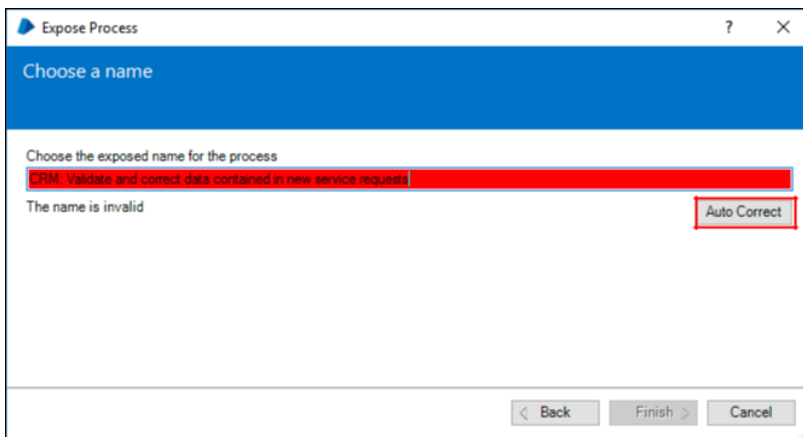
2. 현재 웹 서비스로 노출되지 않은 프로세스 목록이 표시됩니다. 노출될 프로세스를 선택하고 Next 를 클릭하십시오.



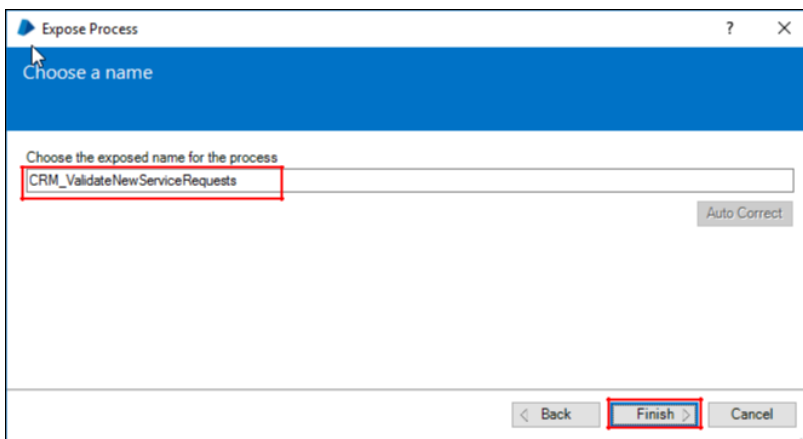
3. 마법사는 프로세스의 이름을 사용하여 웹 서비스에 사용할 적절한 이름을 제안합니다.



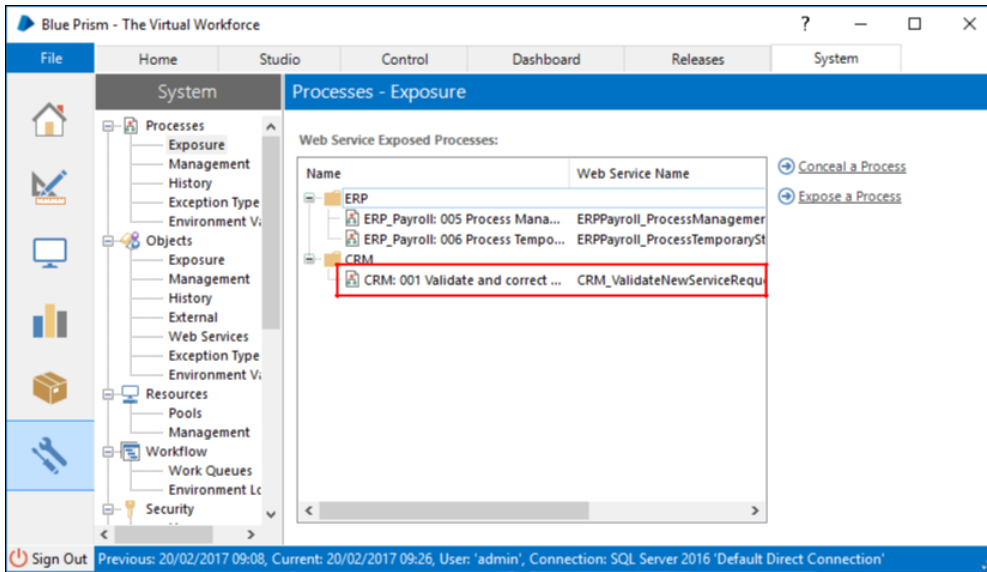
4. 이 시점에서 이름을 변경할 수 있지만 이름이 현재 노출된 Blue Prism 웹 서비스와 중복되거나 잘못된 문자가 감지되면 경고가 표시됩니다. 자동 수정을 클릭하면 잘못된 문자가 제거되고 이름이 고유한지 확인합니다.



5. 이름이 허용되면 Finish 를 클릭합니다.



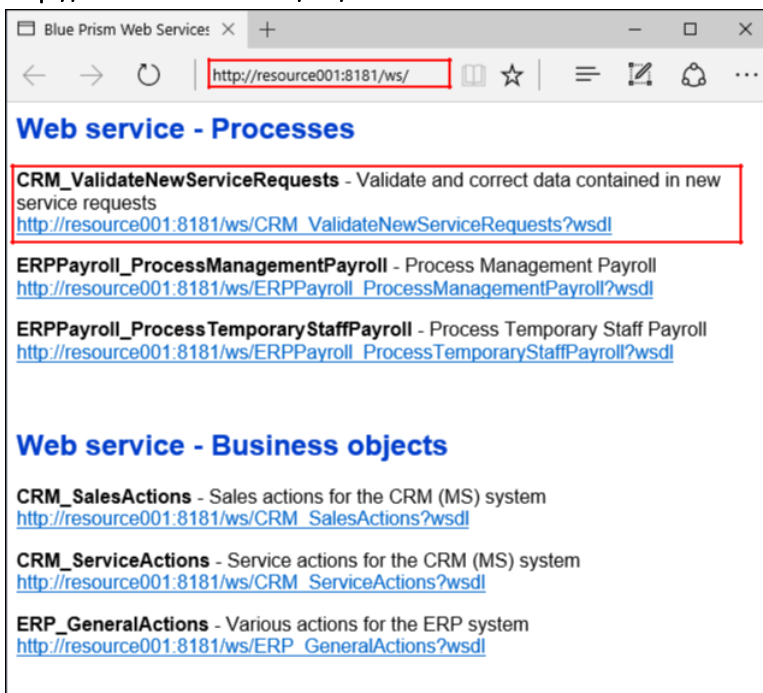
6. 새로 노출된 웹 서비스 이름이 이제 프로세스 이름과 함께 목록에 나타납니다.



7. Blue Prism 에서 노출되는 웹 서비스는 WSDL 의 주소를 제공하는 다음 URL 을 사용하여 확인할 수 있습니다.
[http://\[machinename\]:<port>/ws/](http://[machinename]:<port>/ws/).

이 예에서 컴퓨터 이름은 resource001 이고 기본 포트인 8181 이 사용 중입니다.

<http://resource001:8181/ws/>



Blue Prism 런타임 리소스가 인증서 기반 암호화로 구성된 경우 접두사는 **https** 가 됩니다.

4. Consuming Third-Party Web Services

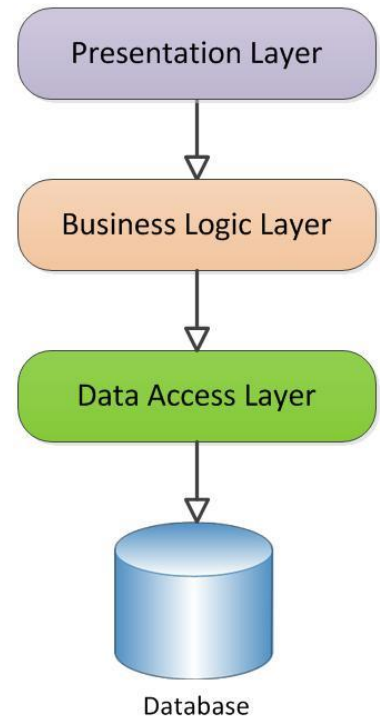
Blue Prism 은 게시되고 액세스 가능한 웹 서비스를 사용하여 타사 응용 프로그램을 자동화하는 기능을 제공합니다.

직관적인 마법사는 사용자가 Blue Prism 을 구성하여 웹 서비스의 수에 제한이 없이 인식할 수 있도록 지원하며 일단 설정이 되면, 구성된 웹 서비스를 Blue Prism 비주얼 디자이너 도구(Object Studio 및 Process Studio) 내에서 사용할 수 있습니다. 이러한 웹 서비스는 코드가 없고 직관적인 방식으로 제공되고 활용됩니다.

GUI(그래픽 사용자 인터페이스)를 사용하여 애플리케이션과 상호 작용하는 일반적인 Blue Prism 자동화와 달리 웹 서비스 통합은 비즈니스 로직 계층(프레젠테이션 계층과 반대)에서 작동하며 추가 복잡성에 취약할 수 있는 전통적인 기술입니다.

웹 서비스 통합 사용을 고려할 때 다음 항목을 검토해야 합니다.

- 애플리케이션이 **Business Logic Layer** 보다 **Presentation Layer** 에서 더 엄격한 데이터 품질 및 유효성 검사 규칙을 적용하는 것이 일반적입니다. 웹 서비스는 **Business Logic Layer** 와 직접 상호 작용하므로 GUI 를 통해서만 제공되는 유효성 검사 로직이 우회될 수 있습니다. 따라서 이러한 검증이 계속 적용되도록 추가 검증 단계를 수동으로 구현해야 할 수 있습니다.
- 웹 서비스는 사용자가 사용자 인터페이스를 통해 수행할 수 있는 것보다 타사 시스템에서 더 많은 수의 작업을 프로세스에 제공할 수 있으므로 추가 IT 거버넌스가 필요할 수 있습니다.
- **Business Logic Layer** 과 직접 연결하면 대상 애플리케이션을 뒷받침하는 플랫폼에 추가적인 부담이 가해질 수 있습니다. 사용량이 많거나 수요가 많은 기간 동안 대상 응용 프로그램이 과도하게 부담되지 않도록 프로세스에 추가 구성을 추가해야 할 수 있습니다.
- 웹 서비스를 게시하는 데 사용할 수 있는 많은 기술, 규칙 및 표준이 있으므로 Blue Prism 내에서 이러한 모든 것을 기본적으로 지원할 수 없습니다. 웹 서비스가 Blue Prism 내에서 지원되는 표준을 따르지 않는 경우 웹 서비스와 인터페이스하기 위해 프로그래밍 기술을 사용해야 합니다.



4.1. Accessing Third-Party Web Services

Blue Prism 은 다음 바인딩 스타일로 SOAP 기반의 타사 웹 서비스를 사용할 수 있습니다.

- RPC/encoded
- document/literal

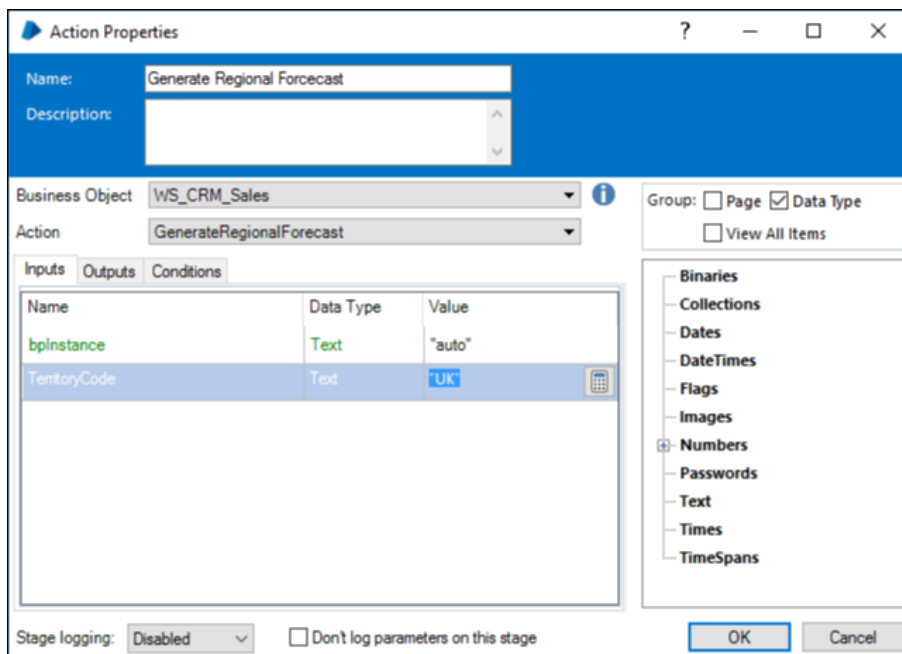
이러한 웹 서비스를 사용할 수 있는 능력은 다음에 의존합니다.

- Blue Prism 내에, 입/출력 웹 서비스 데이터 유형(위에서 설명한대로)에 해당하는 데이터 유형이 있는지
- 웹 서비스 WSDL 이 유효하고 XML 규정을 완전히 포함하는지
- Blue Prism 에 웹 서비스에 대한 적절한 연결이 있는지
- 웹 서비스의 문자 인코딩이 UTF-8 인지
- 접근 권한 관리가 다음 방법 중 하나를 통해 처리되는지:
 - 웹 서비스 호출과 독립적으로(예: IP 보안 기반).
 - 상호 인증서 기반 인증을 사용해서.
 - HTTP 헤더의 일부로 사용자 이름과 비밀번호를 사용해서.

기본적으로 다양한 SOAP 웹 서비스에 대한 지원이 제공되지만 REST, HTTP, JSON 및 Oauth 와 같은 일련의 예제 VBO 를 사용하여 적절한 프로그래밍 경험이 있는 개발자에게 REST 웹 서비스에 대한 지원이 제공됩니다.

Blue Prism 내에서 타사 웹 서비스에 액세스하기 위해 시스템 관리자를 통해 해당 웹 서비스의 SOAP WSDL 을 가져올 수 있습니다. (Walkthrough: Setup a Third-Party Web Service in Blue Prism 를 참조하십시오).

일단 설정되면 기존 Blue Prism Object 에 액세스하는 것처럼 Object Studio 또는 Process Studio 를 통해 웹 서비스를 사용할 수 있습니다.



4.2. Web Service Parameters

Blue Prism 은 예상되는 데이터 유형과 함께 전달할 수 있는 매개 변수를 정의하는 웹 서비스에 매개 변수로 데이터를 전달할 수 있습니다. 다음 웹 서비스 (XSD) 데이터 유형은 Blue Prism 에서 지원됩니다.

Web Service XSD Data Type	Blue Prism Data Type
date	date
dateTime	datetime
boolean	flag
decimal, float, double, pDecimal, integer, nonPositiveInteger, nonNegativeInteger, long, negativeInteger, unsignedLong, int, positiveInteger, unsignedInt, short, unsignedShort, byte, unsignedByte	number
String, normalizedString	text
time	time
duration	timespan
base64binary	binary

소비할 웹 서비스에 위 표에 제시된 유형이 아닌 유형으로 정의된 입력 또는 출력 매개 변수가 있는 경우 이 웹 서비스는 기본적으로 Blue Prism 과 호환되지 않습니다.

4.2.1. Complex Data Types

웹 서비스를 사용할 때 위에 나열된 지원되는 유형으로 구성된 복합 유형(예: 복합 구조 및 배열, 목록)도 사용할 수 있습니다. Blue Prism 내의 컬렉션에 매핑됩니다.

`xsd:all`, `xsd:element`, and `xsd:sequence` 를 포함한 복잡한 데이터 유형 스키마 구성의 하위 집합에 대한 기본 지원이 제공됩니다.

입력 또는 출력 매개 변수를 통해 복잡한 유형(컬렉션)을 노출할 때 컬렉션은 동적으로 정의된 구조가 아닌 미리 정의된 필드를 사용하여 Blue Prism 내에서 설정되어야 합니다.

다형성 데이터 구조를 입력 또는 출력 매개 변수로 사용하는 웹 서비스는 기본적으로 Blue Prism 과 호환되지 않습니다.

데이터 구조가 기본적으로 지원되지 않는 경우 일반적으로 코드 단계와 관련된 고급 기술을 사용하여 이러한 서비스를 사용할 수 있습니다. (See [Consume Web Services using Code Stage Based Business Objects](#) 를 참조하십시오).

4.2.2. WS-Security: Passing Credentials in the SOAP Header

Microsoft WSE (Web Service Enhancements) 3.0 에는 사용자 지정 보안 및 안정적인 메시징과 같은 SOAP 프로토콜에 대한 추가 웹 서비스 사양을 구현하는 클래스 집합이 포함되어 있습니다. 이러한 사양은 일반적으로 WS- 접두사로 표시되며 WS-Security 를 포함합니다.

WSE 3.0 으로 구현된 WS-Security 에서는 웹 서비스 인증 자격 증명(사용자 이름 및 암호)이 SOAP 헤더 내에서 전달되어야 합니다.

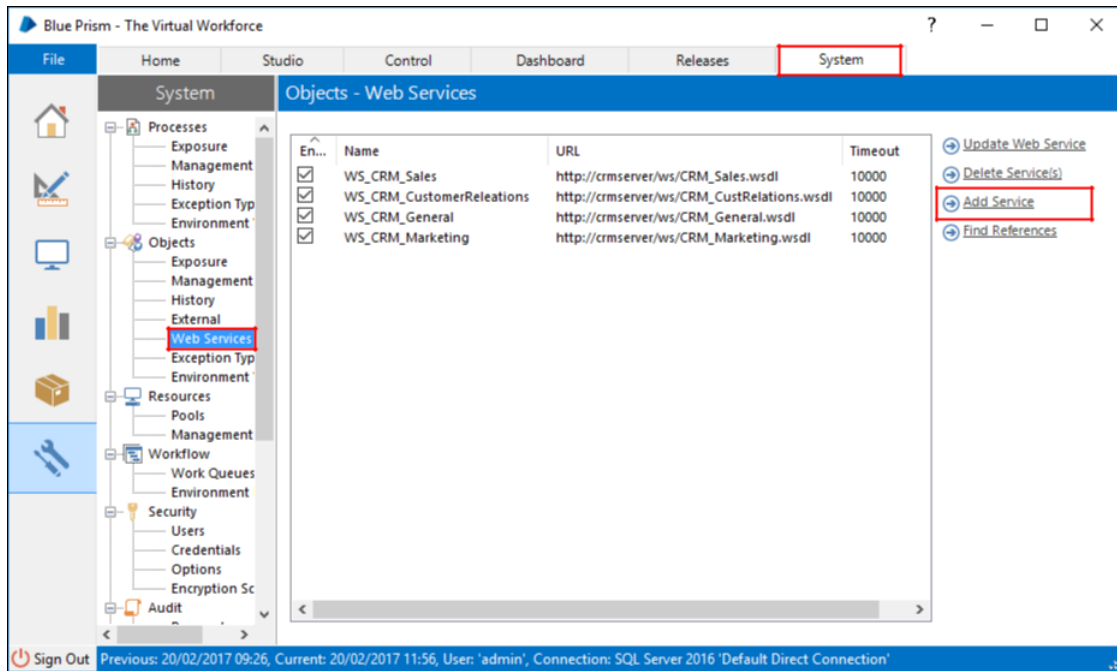
이 구성으로 웹 서비스를 사용할 때 코드 단계를 포함한 고급 기술을 사용해야 합니다. (**Consume Web Services using Code Stage Based Business Objects** 를 참조하십시오).

관련 구성은 코드 단계를 사용하여 수행할 수도 있으며 **MustUnderstand** 와 같은 속성 값 지정 또는 WS-Addressing 헤더 포함 여부 구성을 포함할 수 있습니다.

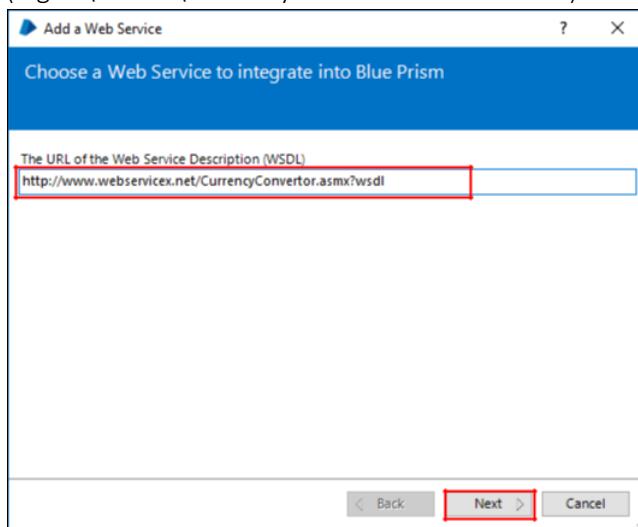
4.3. Walkthrough: Setup a Third-Party Web Service in Blue Prism

이 연습에서는 Blue Prism 내에서 타사 웹 서비스를 사용할 수 있도록하는 데 필요한 단계를 제공합니다. 타사 웹 서비스를 사용/소비하는 방법에 대한 지침은 별도의 연습을 참조하십시오.

1. System 에 들어가서 Objects 영역에서 Web Services 를 선택합니다. Add Service 를 클릭하여 마법사를 시작하십시오.



2. 웹 서비스를 정의하는 WSDL 의 URL 을 입력하고 Next 를 클릭하십시오. WSDL 이 로컬에 저장된 경우 로컬 파일 위치를 입력해도 됩니다. (e.g. c:\WSDLs\CurrencyConversionService.wsdl).



3. 웹 서비스에 필요한 경우, 웹 서비스에 대한 인증 정보(authentication information)를 입력합니다. Next 를 클릭하십시오.

Add a Web Service

Please supply authentication details for the web service.

Web Service Requires Username and password (HTTP Authentication)

Username:

Password:

Web Service Requires client side certificate (SSL Authentication)

Store Location: Store Name: Find Type:

Find Criteria:

Password:

< Back **Next >** Cancel

4. 시간 초과 값을 지정합니다. 요청이 시간 초과되었다고 가정하기 전에 Blue Prism 이 웹 서비스의 응답을 기다리는 시간(밀리 초)입니다. Next 를 클릭하십시오.

Add a Web Service

Please set the timeout to use when interacting with the web service

The timeout to use (in milliseconds) when interacting with the service

< Back **Next >** Cancel

5. Blue Prism 은 제공된 WSDL 에 포함된 정보를 구문 분석합니다. 계속하려면 Next 를 클릭하십시오

Add a Web Service

Importing web service definitions.

Retrieved wsdl from http://www.websvcx.net/CurrencyConvertor.asmx?wsdl

Found service 'CurrencyConverter'

Found service port 'CurrencyConvertorSoap'

Found operation 'ConversionRate'

Found message 'ConversionRateSoapIn'

Found part 'FromCurrency'

Found part 'ToCurrency'

Found message 'ConversionRateSoapOut'

Found part 'ConversionRateResult'

Found service port 'CurrencyConvertorSoap12'

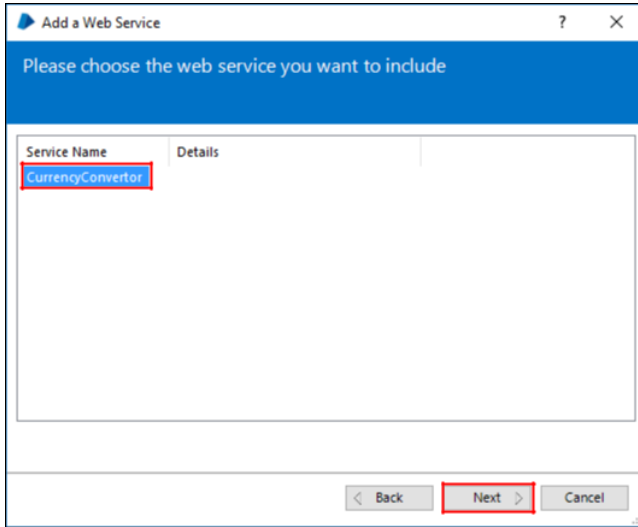
Found service port 'CurrencyConvertorHttpGet'

Found service port 'CurrencyConvertorHttpPost'

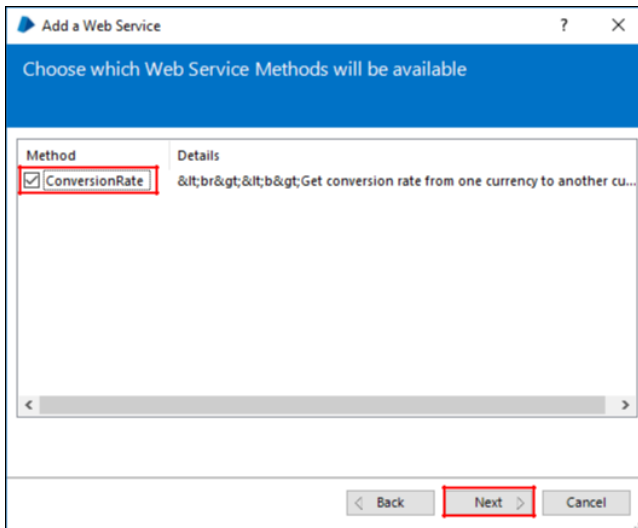
SUCCESS: Retrieved all web service details

< Back **Next >** Cancel

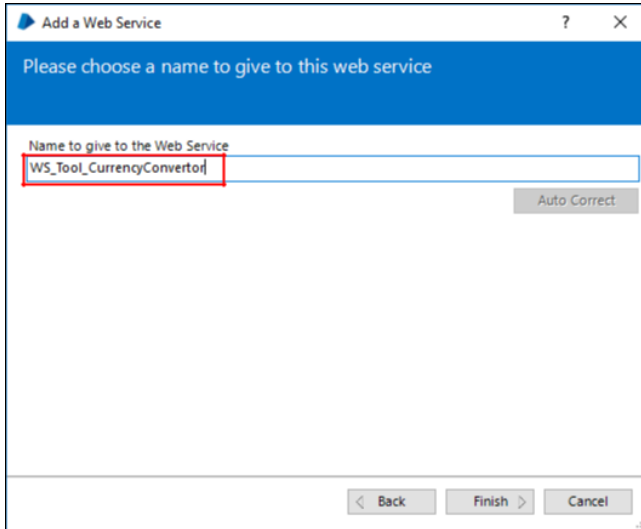
- 6. 사용할 웹 서비스를 선택하고 **Next** 를 클릭하십시오.



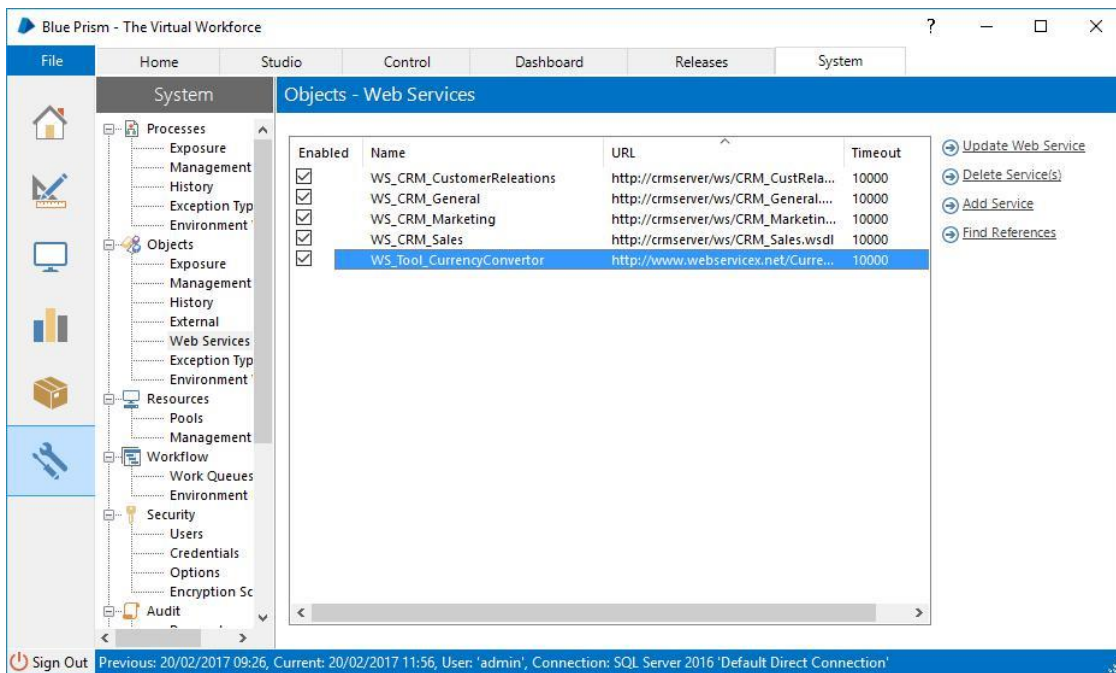
- 7. Blue Prism 내에서 사용할 수 있는 웹 서비스의 **methods** 를 선택하고 **Next** 를 클릭합니다.



8. 마법사는 웹 서비스에 사용할 이름을 제안합니다.
이름이 이전에 노출된 Blue Prism 웹 서비스의 중복이거나 잘못된 문자가 감지되면 경고가 표시됩니다. 자동 수정을 클릭하면 잘못된 문자가 제거되고 이름이 고유한지 확인합니다.



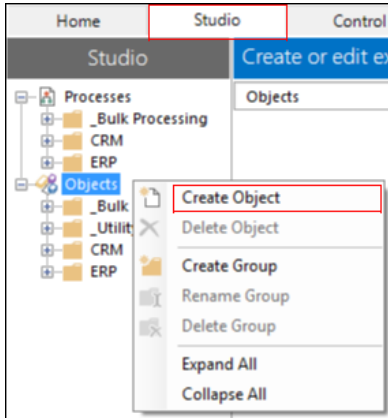
9. 이제 웹 서비스 이름이 활성화되고 사용할 수 있는 웹 서비스 목록에 나타납니다.



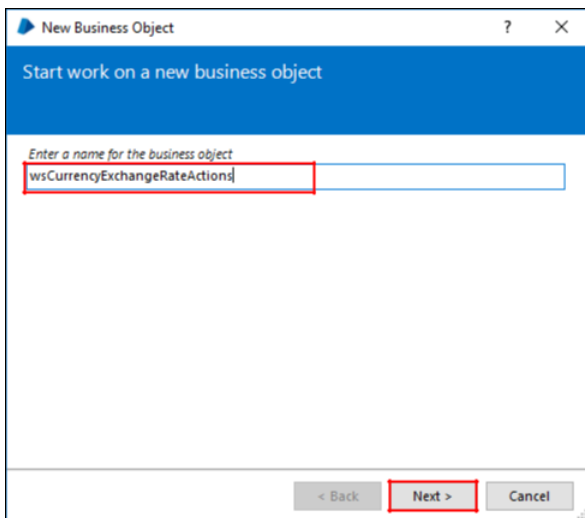
4.4. Walkthrough: Use a Third-Party Web Service in Blue Prism

이 연습에서는 Blue Prism 내에서 제공되는 타사 웹 서비스를 사용하는 방법의 예를 제공합니다. Blue Prism 에서 사용할 수 있도록 타사 웹 서비스를 설정하는 방법에 대한 지침은 별도의 연습을 참조하십시오.

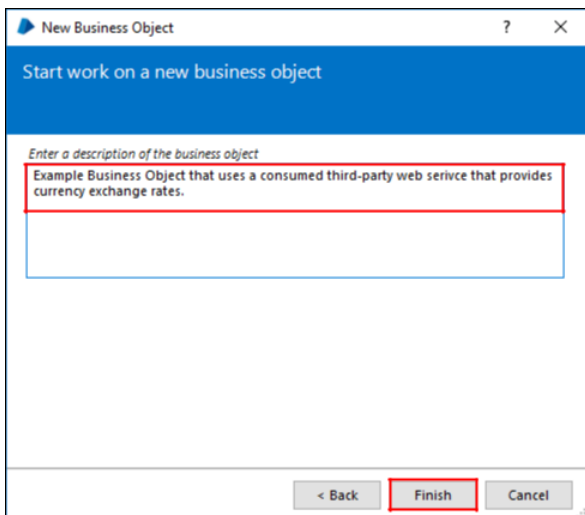
1. Studio 탭에서 마우스 오른쪽 버튼을 클릭하고 Create Object 를 클릭합니다.



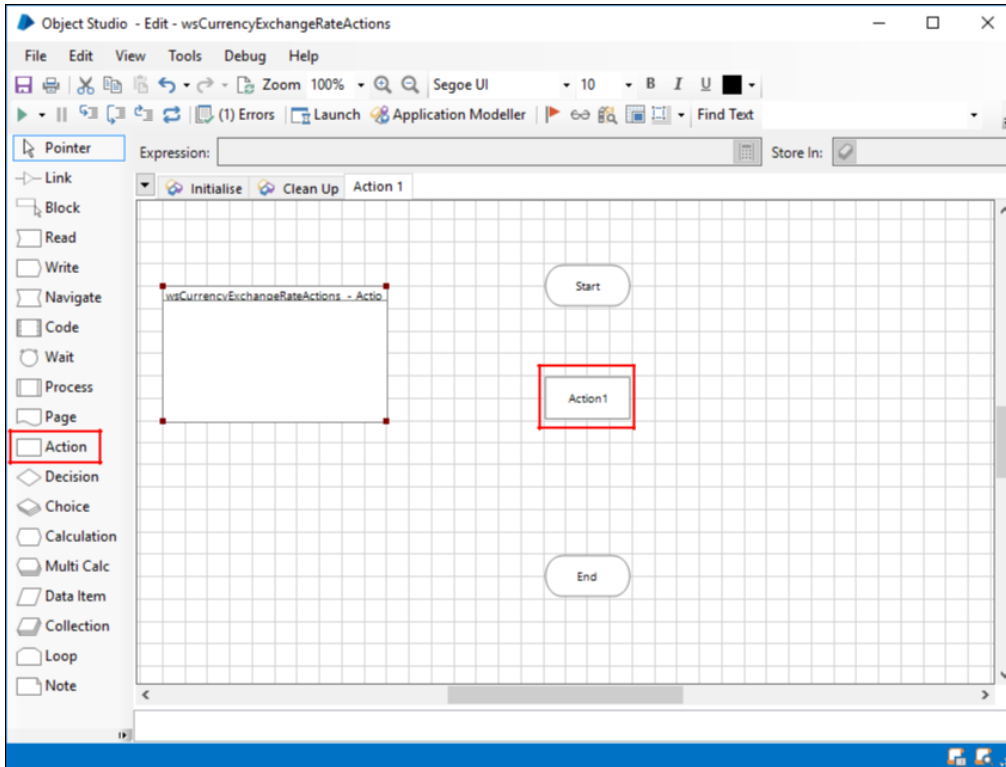
2. 새 비즈니스 오브젝트의 이름을 입력하고 Next 를 클릭하십시오.



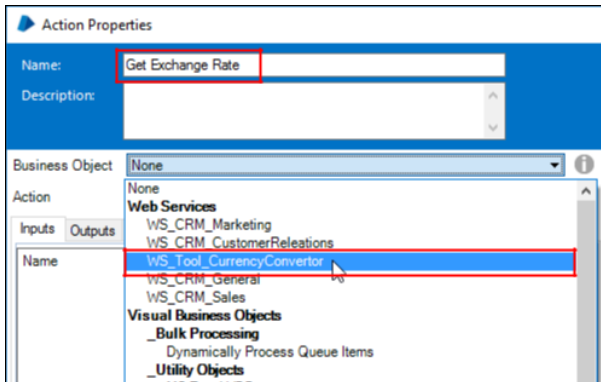
3. 새 비즈니스 오브젝트에 대한 설명을 입력하고 Finish 를 클릭하십시오.



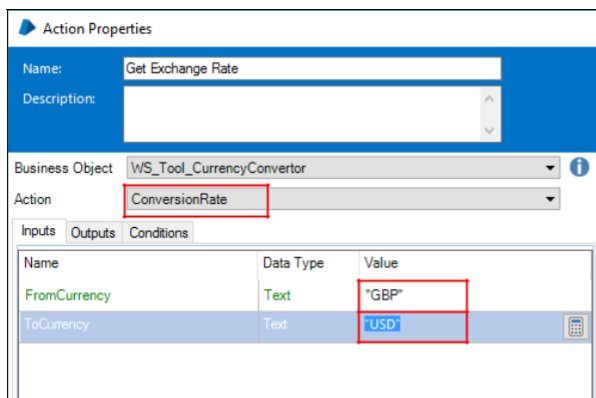
4. 더블 클릭하여 새로 생성된 Business Object 를 열고 Action 1 이라는 페이지를 클릭한 다음 툴바에서 Action 아이콘을 끌어서 Action 항목을 생성합니다.



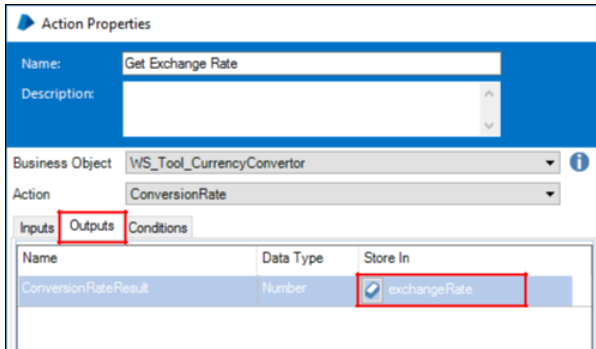
5. Action 항목을 더블 클릭하여 속성을 설정합니다. Name 을 입력하고 Business Object 목록에서 적절한 웹 서비스를 선택합니다.



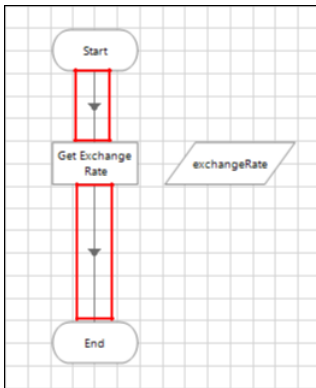
6. 목록에서 적절한 action 을 선택하면 필요한 입력 매개 변수가 표시되어 완료할 수 있습니다.



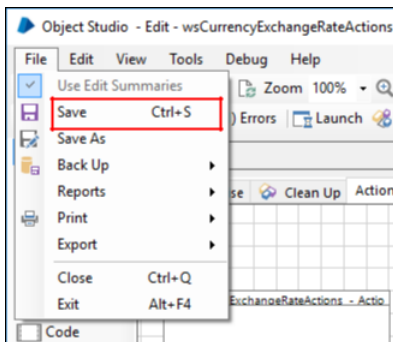
7. **Outputs** 탭을 클릭하고 출력 정보를 저장하는 데 필요한 모든 필수 데이터 항목 또는 컬렉션을 정의합니다. OK 를 클릭하십시오.



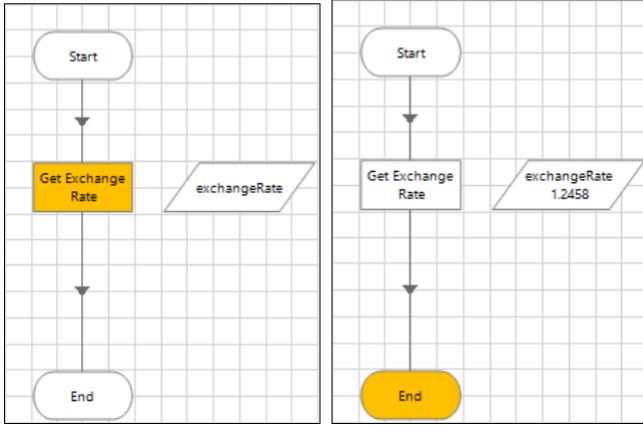
8. 디자이너에서 **Link** 도구를 사용하여 노드를 연결합니다.



9. **File -> Save** 를 클릭하여 비즈니스 오브젝트를 저장하십시오.



- Reset 을 클릭한 다음 Step 단추를 사용하여 노드를 진행하고 출력 매개 변수의 값을 검토하여 웹 서비스가 성공적으로 응답했는지 확인하여 비즈니스 오브젝트를 테스트하십시오.



5. Advanced Topics

5.1. Design Considerations for a Blue Prism Web Service Interface

외부 시스템에서 사용하기 위해 Blue Prism 웹 서비스를 노출하기로 결정할 때 접근 방식이 의도된 요구 사항을 충족하고 Blue Prism 플랫폼의 기능과 일치하는지 확인하기 위해 검토해야 할 여러 설계 고려 사항이 있습니다.

- **Synchronous vs Asynchronous Processing**

요청을 받는 즉시 완전히 서비스되어야 하는 요구 사항이 있는 경우 프로세스의 특성상 종종 다양한 다른 시스템과 상호 작용하고 잠재적으로 복잡한 규칙 및 결정을 적용해야 한다는 점을 고려해야 합니다. 이로 인해 수신 리소스가 비교적 오랜 시간 동안 사용 중이 되고 소수의 시나리오를 제외하고 이 기간 동안 추가 요청을 수락할 수 없게 됩니다.

프로세스의 복잡성으로 인해 처리가 비동기 처리를 위해 Blue Prism 런타임 리소스의 "팀"에 전달될 수 있다면 전체 솔루션에 상당한 이점이 있을 수 있습니다. 이 접근 방식은 용량을 사용할 수 있게 되면 항목을 순서대로 작업하여 향상된 수요 관리를 제공합니다.

또한 비동기 처리는 메시지 대기열, 예외 처리 그리고 네트워크 및 애플리케이션 성능의 변화에 대한 대응과 같이 플랫폼에서 제공하는 기능을 더 잘 활용할 수 있습니다.

- **Processing Volumes**

요청 처리 패턴과 볼륨은 특히 주어진 런타임 리소스 PC 에서 동시 요청을 수신해야 하거나 수요가 급증할 가능성이 있는 경우 웹 서비스 인터페이스의 디자인에 영향을 줍니다.

요청을 수신한 런타임 리소스가 기술적 관점에서 여러 요청을 수신할 수 있는지 여부를 고려해야 합니다. 이것은 다음과 같이 규정됩니다.

- 요청의 특성상 수신된 각 요청에 대해 새 세션을 만들어야 하는지 여부입니다. 이것은 생성되는 세션의 실행 모드에 의해 더욱 영향을 받을 뿐만 아니라 런타임 리소스가 담당해야 하는 기타 작업에 의해서도 영향을 받습니다.
- 플랫폼 라이선스가 최대 수요 기간에 필요할 수 있는 세션 볼륨을 충족하는지 여부.

요구 사항에 따라 웹 서비스 인터페이스는 새 세션을 생성하기 위한 요구 사항과 이후 라이선스에 미치는 영향을 모두 예측하고 제어할 수 있도록 설계할 수 있습니다.

부적절하게 설계된 인터페이스는 요청량이 적을 때에도 가용성 및 성능 제한을 초래할 수 있습니다.

5.1.1. Session Management

Blue Prism 세션은 런타임 리소스가 작업을 수행하고 실행의 런타임 인스턴스를 나타내는 방식의 기본 원칙입니다. 또한 일반적으로 Blue Prism 라이선스가 동시에 실행할 수 있는 세션 수를 제한한다는 점에서 라이선스 제한이 적용됩니다.

인터페이스가 기술적으로 그리고 라이선싱 관점에서 여러 동시 요청을 수락할 수 있는지 여부를 이해하려면 디자인이 세션 관리에 어떤 영향을 미칠 수 있는지와 단일 리소스 PC 가 여러 동시 연결을 수락할 수 있는지 여부를 이해하는 것이 중요합니다.

5.1.1.1. Session Management Design

웹 서비스 요청이 Blue Prism Process 또는 Business Object 로 향하는 지 여부와 자동 기동 사용 여부에 따라 세션이 다르게 관리됩니다.

- Connecting to an **Exposed Process**
 노출된 프로세스에 대한 각 요청은 새로운 세션을 생성해야 하므로 라이선스에 영향을 미칠 수 있는 잠재적으로 많은 수의 세션이 생성됩니다. 또한 수신 런타임 리소스가 여러 개의 동시 활성 세션을 가질 수 있어야 합니다. 일반적으로 이는 기존 세션의 실행 모드와 생성 중인 세션의 실행 모드에 의해 제한됩니다.
- Connecting to a **Business Object using manual-initialization**
 주어진 런타임 리소스가 지정된 시간에 여러 개의 동시 세션을 활성화할 수 있는, 라이선스에 영향을 미칠 수 있는 기능이 필요할 것이며 또한 수신하는 런타임 리소스가 여러 개의 동시 활성 세션을 가질 수 있는 기능을 필요로 합니다.
- Connecting to a **Business Object using auto-initialization**
 Blue Prism 은 연결이 자동 기동을 사용하도록 설정되어 노출된 비즈니스 객체에 대한 요청을 처리하기 위해 세션에 대한 요청 할당을 자동으로 관리합니다. 이는 리소스 당 단일 세션이 잠재적으로 무제한 요청을 처리할 수 있음을 의미합니다. 즉, 새 세션이 지속적으로 필요하지 않기 때문에 라이선스 영향이 별로 없습니다. 리소스가 다른 유형의 처리를 동시에 처리할 것으로 예상되지 않는 한 런타임 리소스에 여러 개의 동시 활성 세션이 있어야 하는 요구 사항은 크게 완화됩니다.

지정된 런타임 리소스에서 추가 동시 세션을 만들 수 있는지 여부는 다음과 같은 근거에 의합니다.

- **Licensing**

일반적으로 Blue Prism 라이선스는 자산 전체에 대해서 동시에 활성화할 수 있는 세션 수를 제한하므로 추가 세션을 생성할 수 있는 충분한 용량이 있어야 합니다.

- **Session Run Mode**

주어진 런타임 리소스의 현재 세션과 새로 요청된 세션의 실행 모드는 생성 가능 여부에 영향을 줍니다. 세션의 실행 모드는 실행 중인 프로세스 또는 Business Object 에서 참조하는 Business Object 에 적용되는 가장 제한적인 설정에 의해 결정됩니다.

사용 가능한 Business Object 실행 모드는 다음과 같습니다.

- **Foreground:** 객체는 주어진 리소스 PC 에서 둘 이상의 활성 인스턴스를 가질 수 없습니다. 이 모드를 사용하면 Business Object 가 백그라운드 Business Object 와 동시에 실행될 수 있습니다.

기존 세션이 포 그라운드로 설정된 경우 포 그라운드 또는 배타적으로 설정된 모든 후속 세션은 원래 세션이 완료될 때까지 거부됩니다. 후속 백그라운드 세션이 허용됩니다.

- **Background:** 이 객체는 동일한 리소스 PC 에서 동시에 실행되는 여러 인스턴스를 지원하도록 설계되었습니다. 이 모드를 사용하면 Business Object 가 포 그라운드 및 백그라운드 비즈니스 오브젝트와 동시에 실행될 수 있습니다.

기존 세션이 백그라운드로 설정된 경우 독점으로 설정된 모든 후속 세션은 원래 세션이 완료될 때까지 거부됩니다. 포 그라운드 세션이 1 개 이하인 경우 후속 백그라운드 또는 포 그라운드 세션이 허용됩니다.

- **Exclusive:** 이 객체는 지정된 리소스 PC 에서 활성 인스턴스를 두 개 이상 가질 수 없으며 다른 비즈니스 객체와 동시에 실행되지 않습니다.

기존 세션이 exclusive 로 설정되어 있으면 원래 세션이 완료될 때까지 모든 후속 세션이 거부됩니다. 마찬가지로 기존 세션의 실행 모드에 관계없이 모든 이전 세션이 완료될 때까지 독점 세션에 대한 모든 후속 요청이 거부됩니다.

주어진 런타임 리소스에서 생성될 수 있는 다른 세션도 고려해야 합니다. 예를 들어 프로세스와 비즈니스 오브젝트의 혼합이 노출되거나 런타임 자원이 임시 또는 예약된 작업을 처리하는데도 사용되는 경우 추가 세션이 필요합니다.

5.1.1.2. Licensing and Session Management Examples

다음 예제에는 세션 관리 및 라이선스 관련 사항을 설명하는 여러 시나리오가 포함되어 있습니다.

Scenario 1

“Runtime Resource 1”은 현재 Foreground 실행 모드에서 예약된 프로세스 “Process A”를 실행하고 있습니다. 빠르게 연속적으로 “Process B” (Foreground 실행 모드에서도 작동)를 대상으로 하는 25 개의 웹 서비스 요청을 받습니다.

- "프로세스 A"(Foreground) 실행을 위해 현재 활성화된 세션이 있습니다.
- “Runtime Resource 1”은 첫 번째 웹 서비스 요청에 대한 세션 생성을 시도하지만 동일한 유형 중 하나가 준비되어 있어서 포 그라운드 세션을 생성할 수 없습니다. 런타임 리소스가 사용 중임을 나타내는 메시지와 함께 요청이 거부됩니다.
- “Runtime Resource 1”은 이후의 각 웹 서비스 요청에 대해 세션을 생성하려고 합니다. 이러한 요청은 원래 포 그라운드 세션 요청이 완료되기 전에 도착하고 요청이 포 그라운드 처리를 위한 것이기 때문에 이러한 요청은 런타임 리소스가 사용 중임을 나타내는 메시지와 함께 거부됩니다.

Total Sessions: 1

Rejected Requests: 25 (excluding rejections due to licensing capacity)

Scenario 2

“Runtime Resource 1”은 현재 Foreground 실행 모드에서 예약된 프로세스 “Process A”를 실행하고 있습니다. 연속적으로 “Process C”(Background 실행 모드에서 작동)를 대상으로 하는 25 개의 웹 서비스 요청을 받습니다.

- "프로세스 A"(Foreground) 실행을 위해 현재 활성화된 세션이 있습니다.
- “Runtime Resource 1”은 첫 번째 웹 서비스 요청에 대한 세션 생성을 시도합니다. 충분한 라이선스 용량이 있다고 가정하면 세션이 성공적으로 생성됩니다.
- “Runtime Resource 1”은 이후의 각 웹 서비스 요청에 대해 세션을 생성하려고 합니다. 충분한 라이선스 용량에 따라 이러한 각 세션이 성공적으로 생성됩니다.

Total Sessions: 26

Rejected Requests: None (excluding rejections due to licensing capacity)

Scenario 3

“Runtime Resource 1”은 현재 Exclusive 실행 모드에서 예정된 프로세스 “Process D”를 실행하고 있습니다. 연속적으로 “Process C” (Background 실행 모드에서 작동)를 대상으로 하는 25 개의 웹 서비스 요청을 받습니다.

- "Process D"(Background) 실행을 위해 현재 활성화된 세션이 있습니다.
- “Runtime Resource 1”은 첫 번째 웹 서비스 요청에 대한 세션 생성을 시도하지만 기존 세션이 Exclusive 모드로 실행되도록 설정되어 있으므로 거부됩니다.
- “Runtime Resource 1”은 이후의 각 웹 서비스 요청에 대해 세션을 생성하려고 합니다. 기존 세션이 Exclusive 모드로 실행되도록 설정되어 있으므로 모두 거부됩니다.

Total Sessions: 1

Rejected Requests: 25 (excluding rejections due to licensing capacity)

Scenario 4

Runtime Resources (1, 2 및 3)는 두 개의 **Business Objects** (A & B)로 구성됩니다.

"Business Object A"에는 **Foreground** 실행 모드가 필요한 반면, "Business Object B"에는 **단독 실행 모드**가 필요합니다.

빠르게 연속적으로, Runtime Resources 1 과 2 는 "Business Object A"를 대상으로 하는 25 개의 웹 서비스 요청을 수신하고 "Runtime Resources 3"은 "Business Object A"를 대상으로 하는 10 개의 요청과 "Business Object B"를 대상으로 하는 10 개의 요청을 받습니다. 요청은 **자동 기동**을 사용해야 함을 지정합니다.

- a) "Runtime Resource 1"은 세션을 만들고 시작하려고 합니다. 이 런타임 리소스에 활성화된 독점 실행 모드 세션이 아직 없고 라이선스 용량이 있기 때문에 성공합니다. 단일 세션은 "Business Object A" (Foreground). 에 대한 25 개의 요청을 모두 처리합니다.
- b) "Runtime Resource 2"는 세션을 만들고 시작하려고 합니다. 실행 모드에 관계없이 이 런타임 리소스에 이미 활성화된 세션이 없고 라이선스 용량이 있기 때문에 성공합니다. 단일 세션은 "Business Object B" (Exclusive)에 대한 25 개의 모든 요청을 처리합니다. 자동 초기화를 사용하면 이 시나리오의 Business Object 가 모든 실행 모드를 가질 수 있습니다.
- c) "Runtime Resource 3"은 세션을 만들고 시작하려고 합니다. 실행 모드에 관계없이 이 런타임 리소스에 아직 활성화된 세션이 없고 라이선스 용량이 있기 때문에 성공합니다. 단일 세션은 "Business Object A" (Foreground)에 대한 10 개의 요청을 처리한 다음 "Business Object B" (Exclusive)에 대한 10 개의 요청을 처리합니다.

Total Sessions: 3

Rejected Requests: None (excluding rejections due to licensing capacity)

이 예제에서 Business Object 웹 서비스를 호출할 때 자동 기동을 사용하면 단일 런타임 리소스에서 처리하기 위해 혼합된 실행 모드(배타적 포함)를 동시에 수신할 수 있습니다.

Scenario 5

두 개의 프로세스(E & F)로 구성된 3 개의 Runtime Resources(1, 2 및 3)가 있는 시나리오를 고려하십시오.
 "프로세스 E"는 백그라운드 모드에서 실행되는 반면 "프로세스 F"에는 배타적 실행 모드가 필요합니다.

빠르게 연속해서, 런타임 리소스 1 과 2 는 각각 "프로세스 E"를 대상으로 하는 25 개의 웹 서비스 요청을 수신하고
 "런타임 리소스 3"은 "프로세스 E"를 대상으로 하는 10 개의 요청과 "프로세스 F"를 대상으로 하는 10 개의 요청을
 받습니다.

- a) "Runtime Resource 1"은 25 개의 요청 각각에 대해 세션을 만들고 시작하려고 합니다. 이 런타임 리소스에 활성화된 단독 실행 모드 세션이 없고 라이선스 용량이 있는 한 성공합니다.
- b) "Runtime Resource 2"는 25 개의 요청 각각에 대해 세션을 만들고 시작하려고 합니다. 이 런타임 리소스에 활성화된 단독 실행 모드 세션이 없고 라이선스 용량이 있는 한 성공합니다.
- c) "Runtime Resource 3"은 10 개의 요청 각각에 대해 세션을 만들고 시작하려고 합니다. 이 런타임 리소스에 활성화된 단독 실행 모드 세션이 없고 라이선스 용량이 있는 한 성공합니다.
- d) "Runtime Resource 3"은 "프로세스 F"에 대한 10 개의 요청 중 첫 번째 세션을 생성하고 시작하려고 시도하지만 런타임 리소스 3 에 이미 다른 활성화된 세션이 있기 때문에 배타적 실행 모드 세션을 허용할 수 없어서 거부됩니다.

Total Sessions: 60

Rejected Requests: 10 (excluding rejections due to licensing capacity)

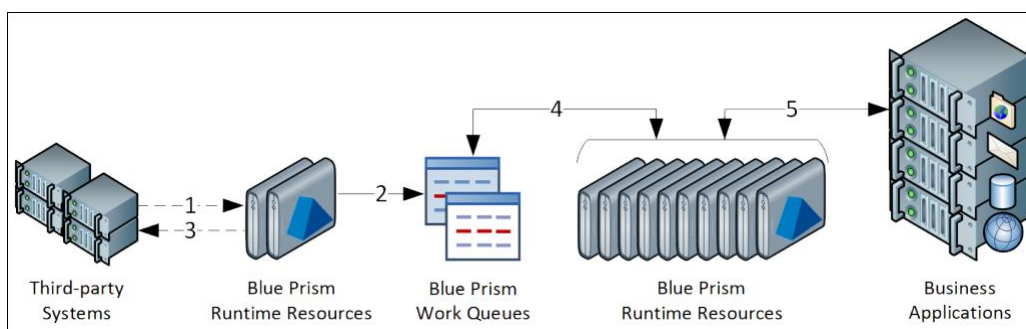
Blue Prism Recommended Approach

다양한 요구 사항에 적합할 수 있는 잠재적으로 많은 접근 방식이 있지만 가장 일반적인 Blue Prism 웹 서비스 인터페이스 디자인에는 다음 기능이 포함됩니다.

- 단순히 데이터를 확인하고 지정된 작업 대기열로 전달하는 Blue Prism Business Object 만 노출합니다. 이상적으로 이러한 비즈니스 오브젝트는 백그라운드 모드에서 실행되도록 설정되지만 포 그라운드 및 독점 실행 모드를 특징으로 하는 시나리오는 잠재적으로 이 구성으로 지원될 수 있습니다. 어떤 경우에는 호출자가 데이터와 대상 작업 대기열을 지정할 수 있도록 하는 매개 변수가 있는 단일 객체 작업을 갖는 것이 적절할 수 있습니다. 다른 시나리오에는 각 대상 작업 대기열에 대한 작업과 함께 Business Object 를 노출하는 것이 포함될 수 있습니다.
- 웹 서비스 요청을 수신하기 위해 소수의 전용 Blue Prism 런타임 리소스를 할당하십시오. 호출자는 시작될 Blue Prism 세션 수를 최소화하고 필요한 경우 단일 세션 내에서 사용되는 혼합 실행 모드를 수용하기 위해 자동 기동을 사용합니다. 다른 처리에 사용되는 것과 같이 단일 런타임 리소스에 여러 세션이 필요한 경우 실행 모드가 충돌하지 않도록 구성해야 합니다.
- 작업 대기열의 항목을 작업하도록 Blue Prism 프로세스를 구성합니다. (이것은 필요에 따라 융통성 있게 할 수 있습니다.)
- 작업 항목의 후속 처리 제어는 Blue Prism 플랫폼에서 관리하는 다른 작업 대기열에 맞춰 조정됩니다. (예: 일정을 사용하여 자동으로 관리하거나 Blue Prism 컨트롤러에 의해 수동으로 관리됩니다). 이를 통해 작업 대기열 정리 전용 런타임 리소스의 수를 수요에 맞게 조정할 수 있습니다.
- 프로세스의 출력은 Blue Prism 에 의해 처리되며 다양한 옵션을 포함할 수 있습니다. 대략적인 옵션 목록에는 대상 시스템 업데이트, 데이터베이스 업데이트, 플랫폼 파일에 결과 로깅, 이메일 보내기, 타사 웹 서비스 호출 등이 포함됩니다.

5.1.1.3. Example

다이어그램은 Blue Prism 웹 서비스를 노출할 때 일반적인 구성을 보여줍니다.



1. 웹 서비스 요청은 Runtime Resource PC 에 의해 수신됩니다. 요청은 Blue Prism Business Object 를 대상으로 하며 처리가 필요한 항목에 대한 정보를 포함합니다.
2. 호출된 작업과 전달된 데이터를 기반으로 정보가 확인되고 작업 항목이 적절한 Blue Prism 작업 대기열에 추가됩니다.
3. 웹 서비스는 작업 대기열에 항목을 성공적으로 추가했는지 여부를 나타내는 응답을 반환합니다.
4. 자동 일정에 따라 Blue Prism 런타임 리소스는 프로세스를 시작하여 작업 대기열을 큐리하고 차례로 각 항목에 대해 공동으로 작업합니다. (Blue Prism 컨트롤러에 의해 수동으로 트리거 될 수도 있습니다)
5. 프로세스 실행의 일부로 필요한 경우 Blue Prism 런타임 리소스는 다양한 비즈니스 애플리케이션과 상호 작용합니다.

5.1.1.4. Advantages

Blue Prism 이 권장하는 접근 방식의 주요 장점은 다음과 같습니다.

- Business Object 작업의 복잡성이 낮기 때문에 실행 모드를 백그라운드 작업으로 설정하여 동시 요청을 수신할 수 있습니다.
- 요구되는 Blue Prism 세션의 수를 최소화하면서 웹 서비스 수요의 높은 변동을 수용합니다. 이를 통해 라이선스의 영향을 예측하고 관리할 수 있습니다.
- 후속 처리에서 작업 대기열 항목의 로깅을 분리하면 처리 용량을 요청 수신 용량과 독립적으로 확장할 수 있습니다. 또한 후속 처리의 비동기 적 특성으로 인해 작업 대기열에 할당된 런타임 리소스의 수를 수요에 따라 확장할 수 있으며 복잡한 결정 및 포괄적인 예외 처리 기능을 활용할 수 있습니다.

5.2. Securing Exposed Blue Prism Web Services

신뢰할 수 있는 보안 로컬 네트워크 외부에 있는 시스템에 Blue Prism 웹 서비스 연결을 제공해야 하는 경우 SSL(Secure Socket Layers)을 사용하여 HTTPS 를 통해 통신하는 것이 바람직할 수 있습니다. 이에 대한 두 가지 일반적인 접근 방식이 있습니다.

- 들어오는 정보 식별 통신에 인증서 기반 암호화를 사용하도록 Blue Prism Runtime Resource 리스너를 구성합니다.
- SSL 을 구현하기 위해 프록시 서버를 배포합니다. 이것은 외부 요청자와 Blue Prism 런타임 리소스 사이의 연결이 됩니다.

5.2.1. Configuring Blue Prism Runtime Resources to use certificate-based encryption

Blue Prism 5.0.18 이상에서는 리소스 수신 포트(기본값 8181)에서 수신된 모든 트래픽에 대해 인증서 기반 암호화를 적용하기 위해 로컬 인증서를 사용하도록 런타임 리소스를 구성할 수 있습니다.

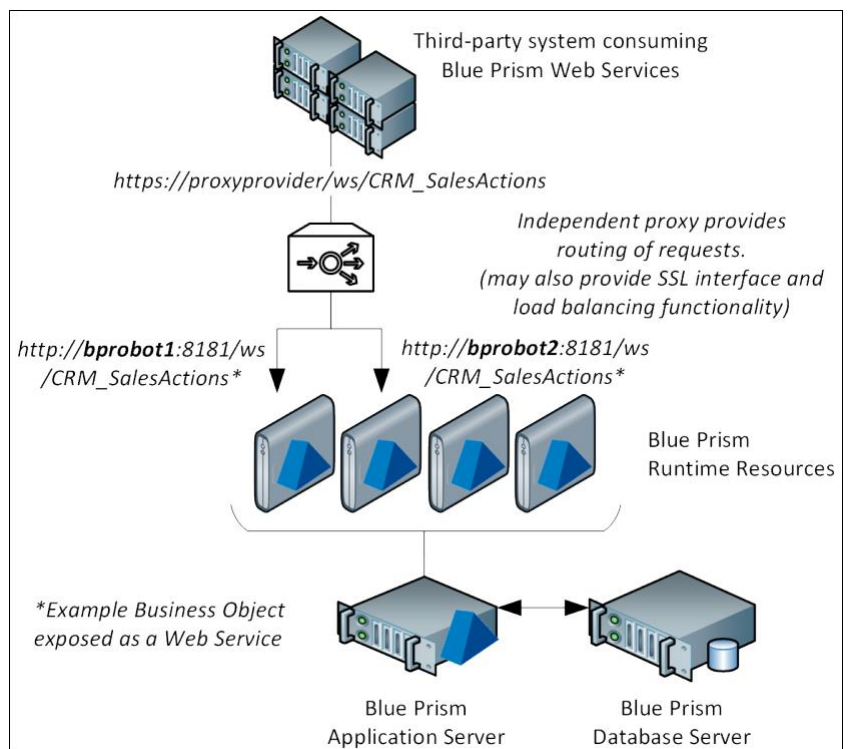
이는 Blue Prism 인터랙티브 클라이언트, Blue Prism 서버(스케줄러) 및 웹 서비스 쿼리의 지침을 포함하여 런타임이 수신한 모든 지침 정보에 영향을 줍니다. 보안 Blue Prism 런타임 리소스에서 호스팅되는 웹 서비스를 사용할 때 URL 은 HTTPS 접두사를 사용합니다. (e.g. <https://bprobot1:8181>)

추가 정보는 [Blue Prism Data Sheet – Securing Network Connectivity](#) 에서 제공됩니다.

5.2.2. Securing Blue Prism Web Services using a proxy

이 구성을 달성하기 위한 한 가지 옵션은 다이어그램에 표시된 대로 프록시 서버를 구현하는 것입니다. 구현된 프록시 서버에 따라 온라인 사용 가능한 런타임 리소스 간에 통신을 분산할 수 있는 로드 밸런싱과 같은 추가 기능을 활용할 수 있는 범위가 있을 수도 있습니다.

이 시나리오에서 프록시는 외부(비 Blue Prism) 트래픽에 대해 마샬링하는 데만 사용됩니다.



5.3. Consume Web Services using Code Stage Based Business Objects

일부 웹 서비스에는 Blue Prism 과 기본적으로 호환되지 않는 기능이 포함되어 있습니다. 이러한 경우 코드 단계를 사용하여 Blue Prism 에서 해석 및 사용을 허용하는 맞춤형 웹 서비스 래퍼를 구축할 수 있습니다.

이 가이드에서는 코드 단계를 기반으로 이러한 시나리오에 대한 웹 서비스 래퍼를 빌드하는 방법을 설명하고 Microsoft 웹 서비스 통합 기술을 사용합니다.

이러한 기술은 성숙하고 자동 생성 코드를 사용하여 다형성 유형 웹 서비스 작업 매개 변수와 같은 기능을 지원합니다. 그러나 이러한 자동 생성 코드는 정적이며 웹 서비스를 변경하려면 재 작업이 필요할 수 있습니다.

WSE 3.0 에 의해 구현된 WS-Security 인증을 활용하는 웹 서비스의 SOAP 헤더에서 자격 증명을 전달하기 위해 코드 단계를 사용하는 방법에 대한 지침을 제공하는 추가 단계가 예제에 포함되어 있습니다. 예제 내의 대체 또는 추가 단계는 WS-Security 구현으로 표시됩니다.

MustUnderstand 속성을 토글하거나 WS-Addressing 헤더를 포함해야 하는지 정의하는 것과 같은 추가 사용자 정의 기능을 제어하기 위해 코드 단계를 추가로 사용자 정의할 수 있지만 이는 이 예제의 범위를 벗어납니다.

5.3.1.1. Pre-requisites

이 가이드는 다음과 같은 기술을 가진 기술 사용자를 대상으로 합니다.

- 웹 서비스에 대한 기본 실무 지식. 특히 WSDL 문서와 익숙하고, 더우기 WS-Security 에 익숙하면 더 유용합니다.
- 객체와 액션 및 객체 내에서 스테이지를 만드는 방법을 포함하여 Blue Prism 제품에 대한 깊은 지식.
- C# 또는 Visual Basic 을 사용하여 Visual Studio .NET 에서 응용 프로그램을 작성할 정도의 높은 이해력.

다음 도구가 필요합니다.

- 프로세스를 생성, 수정, 게시 및 실행할 수 있는 적절한 사용자 계정 권한이 있는 이미 설치된 Blue Prism.
- Visual Studio 2005 or above.

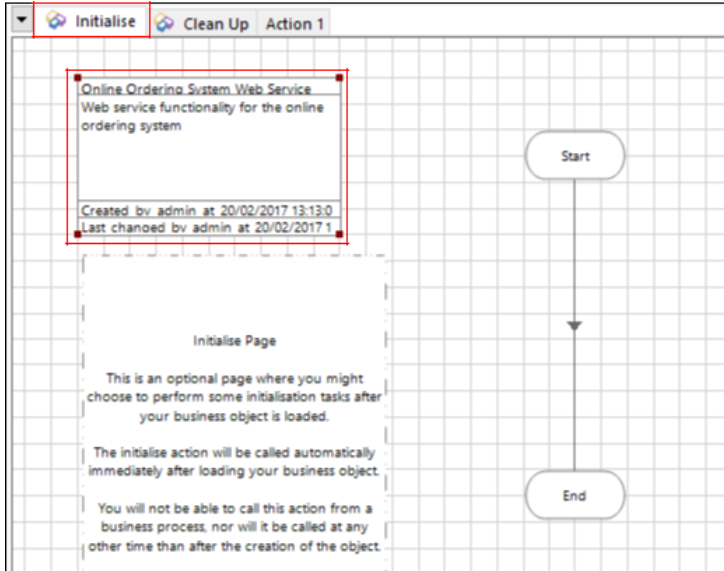
5.3.1.2. Steps

이 가이드에서는 다음 단계를 설명합니다.

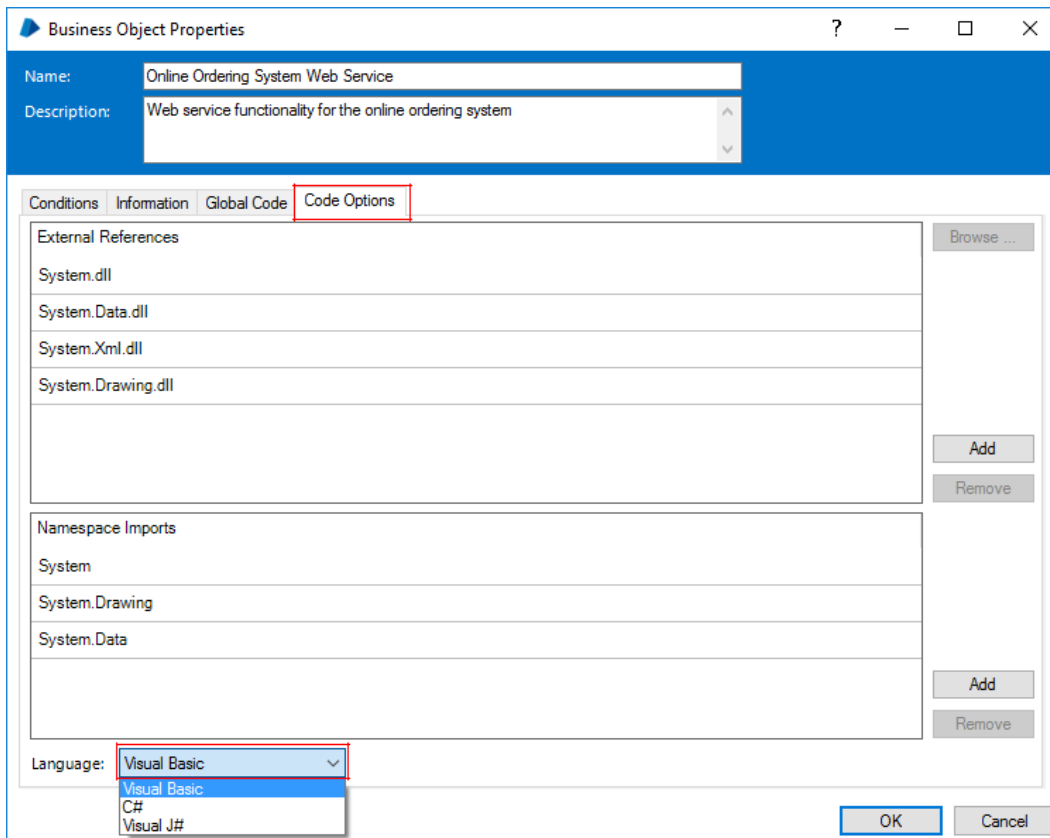
1. Blue Prism 과 웹 서비스 간의 인터페이스를 제공하도록 구성할 새 비즈니스 객체를 생성하고 선호하는 코딩 언어를 선택합니다.
2. 적절한 도구를 사용하여 WSDL 해석을 위한 기본 코드를 자동 생성하십시오.
 - a. **WS-Security Implementation** 예제를 따르는 경우, 각 개발자의 대화형 클라이언트 와 각 런타임 리소스 PC 에 WSE 를 설치해야 합니다.
3. 생성된 코드의 적절한 요소와 필요한 참조 라이브러리를 사용하여 Blue Prism Code Stage 를 설정합니다.
4. 나중에 쉽게 업데이트할 수 있도록 웹 서비스 URL 을 환경 변수로 정의하십시오.
5. Blue Prism 과 웹 서비스 간의 입력 및 출력을 변환하는 웹 서비스 래퍼를 생성합니다.

5.3.2. Generate a new Business Object

1. Blue Prism 내에 새 **비즈니스 객체**를 만듭니다. 이름과 설명을 지정하십시오.
2. 객체가 생성되면 **Initialise**, 라고 표시된 탭을 클릭한 다음 **Page Info** stage (아래 그림 참조)를 **double click** 합니다.



3. 여기에 **Global Code** 및 **Code Options** 탭이 있습니다. **Code Options** 을 클릭하고 원하는 언어를 선택하십시오.



5.3.3. Auto-Generate Code using wsdl.exe or wsewsdl3.exe

일반적으로 Windows SDK 는 Visual Studio 설치의 일부로 설치되며 여기에는 웹 서비스에 대한 래퍼 클래스를 만드는 데 사용할 수 있는 **wsdl.exe** 도구가 포함되어 있습니다.

또는 WSE 3.0 이 설치된 경우 대체 **wsewsdl3.exe** 도구를 사용할 수 있습니다. 이것은 WS-Security 를 구현하고 SOAP 헤더에 자격 증명을 전달해야 하는 웹 서비스를 사용할 때 사용해야 합니다.

생성된 래퍼 클래스는 비즈니스 오브젝트의 Global Code 내에서 사용됩니다.

1. Visual Studio 명령 프롬프트를 시작합니다

일반적으로 시작 메뉴의 관련 **Microsoft Visual Studio** 폴더에서 찾을 수 있습니다.

(예: Programs -> Microsoft Visual Studio -> Visual Studio Tools -> Visual Studio Command Prompt).

2. 웹 서비스 WSDL 에 대한 명령 프롬프트를 통해 wsdl.exe 또는 wsewsdl3.exe 도구를 사용하십시오.

wsdl.exe

WS-Security 를 구현하지 않고 SOAP 헤더에 자격 증명을 전달할 필요가 없는 웹 서비스에 적합합니다.

```
wsdl.exe /language:[VB|CS] /out:[File Location] [Address of WSDL]
```

VB.NET

```
wsdl.exe /language:VB /out:"%TEMP%" http://example.com/wsService?wsdl
```

C#

```
wsdl.exe /language:CS /out:"%TEMP%" http://example.com/wsService?wsdl
```

WS-Security Implementation:

WS-Security 추가 예제를 따를 때에는 **wsewsdl3.exe** 도구를 사용해야 합니다.

wsewsdl3.exe

WS-Security 를 구현하고 SOAP 헤더에 자격 증명을 전달해야 하는 웹 서비스에 적합합니다.

```
wsewsdl3.exe /language:[VB|CS] /type:webClient /out:[File Location] [Address of WSDL]
```

VB.NET

```
wsewsdl3.exe /language:VB /type:webClient /out:"%TEMP%" http://example.com/wsService?wsdl
```

C#

```
wsewsdl3.exe /language:CS /type:webClient /out:"%TEMP%" http://example.com/wsService?wsdl
```

도구의 출력은 아래 메시지와 유사해야 하며 생성된 출력 파일의 특정 위치를 나타냅니다.

```
Microsoft (R) Web Services Description Language Utility
[Microsoft (R) .NET Framework, Version 2.0.50727.42]
Copyright (C) Microsoft Corporation. All rights reserved.
Writing file 'C:\Temp\Service1.vb'.
```

4. 파일 위치로 이동하고 적절한 편집기를 사용하여 엽니다.

5.3.4. Copy the relevant code to the Business Object

wSDL.exe 도구에 의해 생성된 코드에는 이전에 작성된 Business Object 에 복사되는 나머지 코드에 앞서 제거해야 하는 헤더가 포함되어 있습니다.

버릴 수 있는 헤더는 다음과 유사합니다.

VB.NET

```

'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:2.0.50727.5466
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

Option Strict Off
Option Explicit On

Imports System
Imports System.ComponentModel
Imports System.Diagnostics
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Xml.Serialization
'

' This source code was auto-generated by wsdl, Version=2.0.50727.42.
'
'''<remarks/>

```

C#

```

//-----
// <auto-generated>
//   This code was generated by a tool.
//   Runtime Version:2.0.50727.5466
//
//   Changes to this file may cause incorrect behavior and will be lost if
//   the code is regenerated.
// </auto-generated>
//-----

using System.ComponentModel;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml.Serialization;

//
// This source code was auto-generated by wsdl, Version=2.0.50727.42.
//

```

나머지 코드를 Business Object 의 Global Code 탭에 수동으로 복사하십시오.
 이때 Check Code 버튼을 누르지 마십시오.

클래스 이름을 기록해 둡니다(아래 코드 예제에서는 Service1 입니다).

VB . NET

```

1 <System.CodeDom.Compiler.GeneratedCodeAttribute("wsdl", "2.0.50727.42"
2 System.Diagnostics.DebuggerStepThroughAttribute(), _
3 System.ComponentModel.DesignerCategoryAttribute("code"), _
4 System.Web.Services.WebServiceBindingAttribute(Name="Service1Soap",
5 Partial Public Class Service1
6 Inherits System.Web.Services.Protocols.SoapHttpClientProtocol
7
8 Private GetAllMemberValuesOperationCompleted As System.Threading.S
9
10 '''<remarks/>
11 Public Sub New()
12 MyBase.New
13 Me.Url = "http://localhost:49220/Service1.asmx"
14 End Sub
15
16 '''<remarks/>
17 Public Event GetAllMemberValuesCompleted As GetAllMemberValuesComp
18
19 '''<remarks/>
    
```

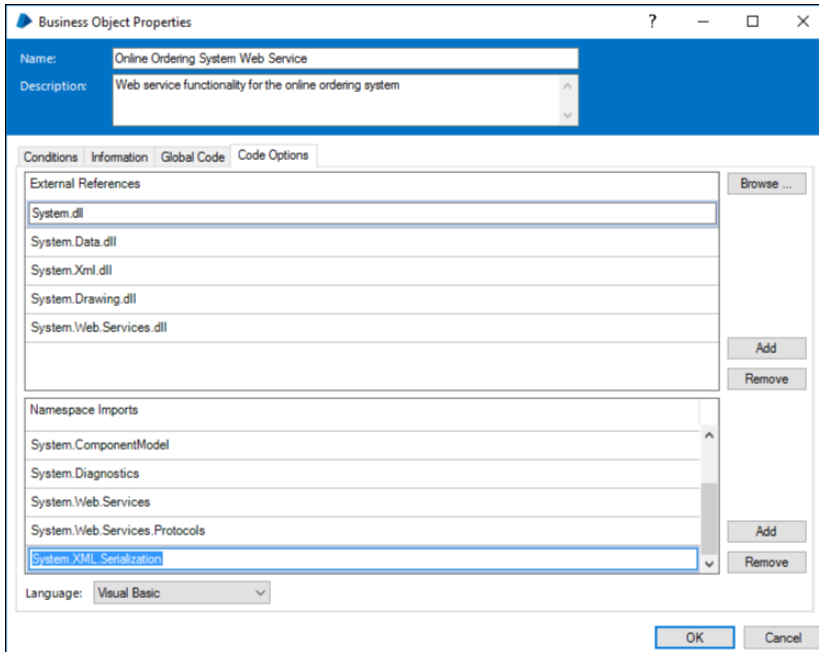
C#

```

1 [System.CodeDom.Compiler.GeneratedCodeAttribute("wsdl", "2.0.50727.42")]
2 [System.Diagnostics.DebuggerStepThroughAttribute()]
3 [System.ComponentModel.DesignerCategoryAttribute("code")]
4 [System.Web.Services.WebServiceBindingAttribute(Name="Service1Soap", Name=
5 public partial class Service1 : System.Web.Services.Protocols.SoapHttpClie
6
7 private System.Threading.SendOrPostCallback HelloWorldOperationComple
8
9 /// <remarks/>
10 public Service1() {
11     this.Url = "http://localhost:1232/Service1.asmx";
12 }
13
14 /// <remarks/>
15 public event HelloWorldCompletedEventHandler HelloWorldCompleted;
16
17 /// <remarks/>
18 [System.Web.Services.Protocols.SoapDocumentMethodAttribute("http://ter
19 public string HelloWorld() {
20     object[] results = this.Invoke("HelloWorld", new object[0]);
    
```

5.3.5. Add the necessary External Library References

1. Code Options 탭을 클릭하여 External References 와 Namespace Imports 에 다음과 같은 것을 추가합니다.



VB.NET

```
External References
System.Web.Services.dll

Namespace Imports
System.ComponentModel
System.Diagnostics
System.Web.Services
System.Web.Services.Protocols
System.Xml.Serialization
```

C#

```
External References
System.dll
System.Web.Services.dll

Namespace Imports
System.ComponentModel
System.Diagnostics
System.Web.Services
System.Web.Services.Protocols
System.Xml.Serialization
```

WS-Security Implementation

WS-Security 를 따를 때에는 다음과 같은 추가 코드 옵션이 필요합니다.

```
External References
Microsoft.Web.Services3.dll

Namespace Imports
Microsoft.Web.Services3.Design
Microsoft.Web.Services3.Messaging
Microsoft.Web.Services3
Microsoft.Web.Services3.Security.Tokens
```

2. Global Code 탭을 선택하고 Check Code 을 클릭하십시오. 오류가 보고되면 계속하기 전에 해결해야 합니다.

5.3.6. Setup the Global Code to use an Environment Variable for the URL

웹 서비스 URL 의 향후 수정을 허용하기 위해 Service1 클래스의 생성자는 Blue Prism 환경 변수에 지정된 값을 사용하도록 수정됩니다.

1. 비즈니스 오브젝트의 Global Code 내에서 기존 코드 맨 위에 멤버 변수를 추가하십시오.

VB.NET

```
Private mWebService As Service1
```

C#

```
private Service1 mWebService;
```

2. 코드 내에서 Service1 클래스 constructor 를 찾고 URL 문자열 매개 변수를 추가합니다. 일반적으로 다음을 검색하여 찾습니다.

VB.NET

```
Public Sub New ()
```

C#

```
public Service1()
```

수정된 생성자는 다음과 유사해야 합니다.

VB.NET

```
Public Sub New(ByVal url As String)
    MyBase.New()
    Me.Url = url
End Sub
```

C#

```
public Service1(string url) : base()
{
    this.Url = url;
}
```

3. Blue Prism 시스템 관리자를 사용하여 새 환경 변수를 만듭니다(프로세스 영역->환경 변수). **add variable** 링크를 클릭하고 원래 코드에 표시된 대로 환경 변수에 이름, 텍스트 유형 및 URL 값을 제공하십시오.
4. Business Object 의 초기화 페이지에서 새로 생성된 환경 변수를 사용하는 데이터 항목을 생성합니다. 데이터 항목의 이름은 환경 변수의 이름과 정확히 일치해야 합니다. 데이터 항목 속성 내의 노출 유형 드롭 다운에서 환경 변수를 선택합니다.
5. 또한 초기화 페이지 내에서 텍스트 유형의 'url'이라는 단일 입력이 있는 코드 단계를 추가합니다. 이는 Blue Prism 환경 변수를 사용하여 채워집니다. 이 코드 단계에 추가할 코드는 다음과 같습니다.

VB.NET

```
mWebService = New Service1(Url)
```

C#

```
mWebService = new Service1(Url);
```

코드 단계는 초기화 페이지의 시작 및 종료 단계에 연결되어야 합니다.

5.3.7. Generate a Wrapper Code Stage

Blue Prism 프로세스가 웹 서비스를 사용하도록 허용하려면 코드 단계 래퍼가 필요합니다. 여기에는 **비즈니스 객체** 내에서 새 Action 을 만드는 작업이 포함됩니다.

래퍼 코드 단계를 작성하려면 맞춤형 구현이 필요하며 호출해야 하는 웹 서비스의 작업에 따라 다릅니다. 가이드의 이 부분에서는 달성해야 할 사항에 대한 개요를 제공하지만 웹 서비스 작업의 특정 요구 사항을 충족하도록 예제를 수정해야 합니다.

래퍼 코드 단계를 생성하는 데 필요한 작업은 다음과 같습니다.

1. Business Object 내에서 새 Action 페이지 만들기
2. 웹 서비스 액션에 필요한 매개 변수를 기반으로 오브젝트 Action 에 대한 적절한 입력 및 출력 매개 변수를 정의하십시오.

Blue Prism 은 데이터 유형을 기본 .NET 데이터 유형으로 코드 단계에 제공합니다. 예 :

- *Text* 는 *System.String* 으로 코드 단계에 표시됩니다.
 - *Number* 는 *System.Decimal* 로 코드 단계에 표시됩니다.
 - A *Collection* 데이터 유형은 *System.Data.DataTable* 로 코드 단계에 제공됩니다. 단일 행 컬렉션을 사용하여 이름 값 쌍을 나타낼 수 있으며 이를 클래스의 멤버 속성에 매핑할 수 있습니다. 단일 필드가 있는 컬렉션을 사용하여 배열 또는 목록 데이터를 나타낼 수 있습니다. 중첩된 컬렉션을 사용하여 데이터 구조와 같은 계층적 또는 트리를 나타낼 수 있습니다.
3. 작업에 코드 단계를 추가하고 시작 및 종료 단계에 연결합니다. 그런 다음 코드 단계에 대한 입력 및 출력 매개 변수를 작업의 입력 및 출력으로 설정한 것과 동일하게 설정해야 합니다.
 4. 코드 단계 자체 내에서 기본 개요로서 코드는 최소한 다음을 수행해야 합니다.
 - a. 입력 매개 변수를 웹 서비스 조치 매개 변수로 변환하십시오.
 - b. 웹 서비스 조치를 호출하십시오.
 - c. 웹 서비스 호출의 결과를 출력 매개 변수로 변환합니다.
 5. 다음 예에 대해서 다음과 같이 주의해야 합니다.
 - a. 모든 행이 처리되도록 들어오는 컬렉션을 처리하는 루프가 구현되었습니다. 컬렉션에 단일 행이 포함될 것으로 예상되는 경우에도 이러한 방식으로 컬렉션을 읽는 것이 좋습니다.
 - b. 출력 데이터 테이블에 지정된 열은 Blue Prism 객체 내의 컬렉션에 정의된 필드 이름과 일치해야 합니다.

필요한 코드의 예는 다음과 같습니다. 이 코드는 예제로 제공되며 수정하지 않고 컴파일할 수 없습니다.

자주색 주석으로 캡슐화된 들여쓰기 된 코드는 WS-Security Implementation 예제를 따르는 경우에만 필요합니다.

VB.NET

```
Dim q As New QueryByName() 'Replace QueryByName with the correct class name
'Example 1 Collection input
Dim inputDataTable As System.Data.DataTable = input1
For Each inRow As System.Data.DataRow In
    inputDataTable.Rows q.Value = inRow("field1")
Next
'Example 2 Number Input
Dim inputNumber As Integer = input2
q.NumberValue = inputNumber
'Example 3 Text Input
Dim inputText As String = input3
q.TextValue = inputNumber
'...

'Example of Invoking the web service action
Dim r As BaseResponse
```

```
'Content required for WS-Security Implement: Start
Dim token As New UsernameToken("webSvcUser", "Pa55word", PasswordOption.SendPlainText)
mWebService.SetClientCredential(Of UsernameToken)(token)
Dim webServiceClientPolicy As New Policy()
webServiceClientPolicy.Assertions.Add(New UsernameOverTransportAssertion())
mWebService.SetPolicy(webServiceClientPolicy)
'Content required for WS-Security Implement: END
```

```
r = mWebService.retrieveValuationsBy(q)

'Example of converting the output of the web
'service action into a datatable for storing
'in the output collection
Dim outputDataTable As New System.Data.DataTable
outputDataTable.Columns.Add("field1")
Dim outRow As System.Data.DataRow = outputDataTable.Rows.Add()
outRow("field1") = r.Value

output1 = outputDataTable
'...
```

C#

```
QueryByName q = new QueryByName(); //Replace QueryByName with correct class name

//Example 1 Collection input
System.Data.DataTable inputDataTable = input1;
foreach (System.Data.DataRow inRow in inputDataTable.Rows)
    { q.Value = inRow["field1"];
}

//Example 2 Number Input
int inputNumber = input2;
q.NumberValue = inputNumber;

//Example 3 Text Input
string inputText = input3;
q.TextValue = inputNumber;
//...

//Example of Invoking the web service action
BaseResponse r = default(BaseResponse);

//Content required for WS-Security Implement: Start
UsernameToken token = new UsernameToken("webSvcUser", "Pa55word", PasswordOption.SendPlainText);
mWebService.SetClientCredential<UsernameToken>(token);
Policy webServiceClientPolicy = new Policy();
webServiceClientPolicy.Assertions.Add(new UsernameOverTransportAssertion());
mWebService.SetPolicy(webServiceClientPolicy);
//Content required for WS-Security Implement: END
r = mWebService.retrieveValuationsBy(q);

//Example of converting the output of the web
//service action into a datatable for storing
//in the output collection
System.Data.DataTable outputDataTable = new System.Data.DataTable();
outputDataTable.Columns.Add("field1");
System.Data.DataRow outRow = outputDataTable.Rows.Add();
outRow["field1"] = r.Value;
output1 = outputDataTable;
//...
```

5.4. Consume an Exposed Blue Prism Web Service using Visual Studio

이 가이드는 Blue Prism 에서 노출된 웹 서비스를 사용할 수 있는 Visual Studio 를 사용하여 독립적인 웹 서비스 클라이언트를 구축하는 방법을 보여줍니다.

제공된 예제 구성 요소의 사용을 포함하는 모두 작동 예제입니다.

5.4.1.1. Pre-requisites

이 가이드는 다음과 같은 기술을 가진 기술 사용자를 대상으로 합니다.

- 웹 서비스에 대한 기본 실무 지식. 특히 WSDL 문서에 익숙하면 유용합니다.
- Business Objects 및/또는 프로세스를 웹 서비스로 노출하는 방법을 포함하여 Blue Prism 제품에 대한 깊은 지식.
- C# 또는 Visual Basic .NET 을 사용하여 Visual Studio 개발에 대한 깊은 이해. 이 예에서는 VB .NET 을 사용합니다.

다음 도구가 필요합니다.

- 프로세스를 생성, 수정, 게시 및 실행할 수 있는 적절한 사용자 계정 권한이 있고 운영 중인 Blue Prism 설치.
- 샘플 Blue Prism 비즈니스 객체 및 프로세스.
- Visual Studio 2005 or above.

5.4.1.2. Steps

이 가이드에서는 다음 단계를 설명합니다.

1. 샘플 비즈니스 오브젝트 및 프로세스 다운로드 및 가져오기
2. 비즈니스 오브젝트 및 프로세스를 웹 서비스로 노출
3. 적절한 라이브러리 참조를 사용하여 새 Visual Studio 콘솔 응용 프로그램 프로젝트를 만듭니다.
4. 노출된 웹 서비스에 링크하는 웹 참조를 추가하십시오.
5. 다음을 위한 간단한 코드 생성하십시오:
 - a. Business Object 기반 웹 서비스와 상호 작용합니다 (수동 및 자동 초기화 사용).
 - b. 프로세스 기반 웹 서비스와 상호 작용합니다.
 - c. 애플리케이션이 실행될 때 위의 상호 작용을 트리거합니다.

5.4.2. Download and Import the Sample Business Object and Process

포털을 통해 리소스 zip 파일을 사용할 수 있습니다. 다음과 같은 내용을 포함하고 있습니다.

- A sample Blue Prism Business Object
- A sample Blue Prism Process
- Example Visual Studio Project (VB.NET) of a Web Service Client

포털에서 다음 파일을 검색하십시오: **Resources_Consuming an Exposed Blue Prism Web Service.zip**

1. **Blue Prism** 을 실행하고 및 로그인하십시오.
2. **File** 메뉴에서 **Import** 마법사를 사용하여 샘플 **Business Object** 와 **Process** 를 각각 가져옵니다.
 - a. BPA Object – NumericOperations.xml
 - b. BPA Process – FavouriteFruitLookup.xml

5.4.3. Expose the Sample Business Object and Process as Web Services

1. 다음 위치를 통해 액세스되는 마법사를 사용하여 비즈니스 오브젝트 및 프로세스를 웹 서비스로 노출합니다. 이 예제의 목적을 위해 노출된 각 웹 서비스의 이름은 아래에 명시된 이름과 일치해야 합니다.

	Wizard Location	Web Service Name
Business Object	System Manager -> Business Objects -> Process Exposure	NumericOperations
Process	System Manager -> Processes -> Process Exposure	FavouriteFruitLookup

비즈니스 오브젝트를 웹 서비스로 노출하고; 프로세스를 웹 서비스로 노출하는 방법에 대한 전체 연습을 사용할 수 있습니다.

2. 다음 URL 로 이동하여 나중에 사용할 각 웹 서비스에 대한 WSDL 의 URL 을 검색합니다.
[http://\[machine name\]:\[port\]/ws/](http://[machine name]:[port]/ws/)

여기서 **[machine name]**은 Blue Prism 런타임 리소스 중 하나의 이름이고, **[port]**는 Blue Prism 이 구성되어 정의된 포트입니다.

E.g. <http://resource001:8181/ws/>

Blue Prism 런타임 리소스가 인증서 기반 암호화로 구성된 경우, 접두사는 **https** 가 됩니다.

Blue Prism Web Services × +

← → ↻ | http://resource001:8181/ws/ | 📖 ☆ | ☰ 🗒️ 🔔 ⋮

Web service - Processes

CRM_ValidateNewServiceRequests - Validate and correct data contained in new service requests
http://resource001:8181/ws/CRM_ValidateNewServiceRequests?wsdl

ERPPayroll_ProcessManagementPayroll - Process Management Payroll
http://resource001:8181/ws/ERPPayroll_ProcessManagementPayroll?wsdl

ERPPayroll_ProcessTemporaryStaffPayroll - Process Temporary Staff Payroll
http://resource001:8181/ws/ERPPayroll_ProcessTemporaryStaffPayroll?wsdl

FavouriteFruitLookup - Provides a dictionary service for the favourite fruit of each of the Seven Dwarfs from the fictional tale of snow white.
<http://resource001:8181/ws/FavouriteFruitLookup?wsdl>

Web service - Business objects

CRM_SalesActions - Sales actions for the CRM (MS) system
http://resource001:8181/ws/CRM_SalesActions?wsdl

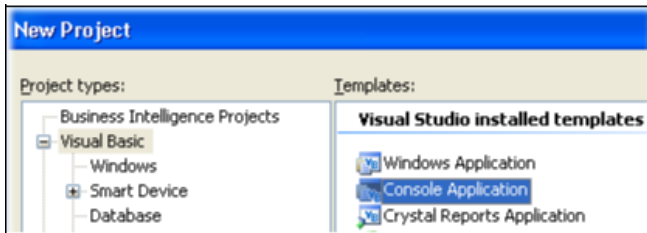
CRM_ServiceActions - Service actions for the CRM (MS) system
http://resource001:8181/ws/CRM_ServiceActions?wsdl

ERP_GeneralActions - Various actions for the ERP system
http://resource001:8181/ws/ERP_GeneralActions?wsdl

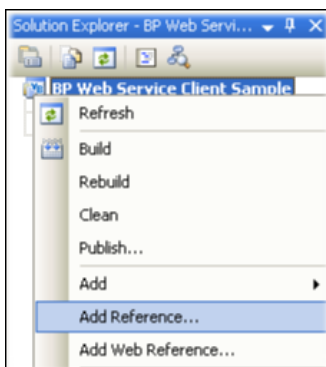
NumericOperations - Provides basic numeric operations such as addition and subtraction
<http://resource001:8181/ws/NumericOperations?wsdl>

5.4.4. Use Visual Studio to create a new Console Application

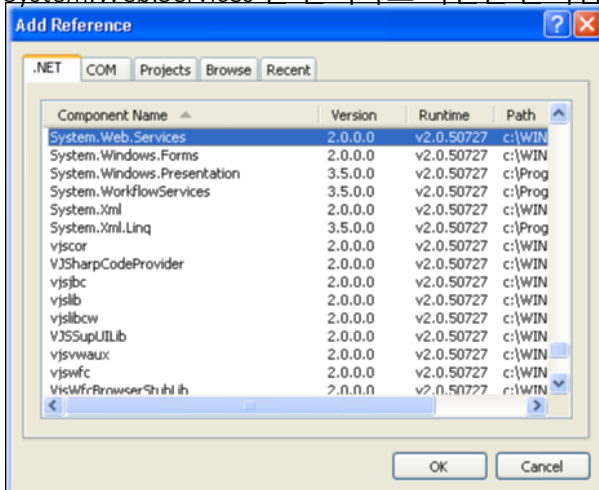
1. BP Web Service Client Sample 이라는 새 Visual Basic Console Application 을 만듭니다.



2. 솔루션 탐색기(solution explorer)에서 프로젝트 이름을 마우스 오른쪽 단추로 클릭하고, Add Reference 를 선택하여 필요한 Library Reference 를 추가하십시오.

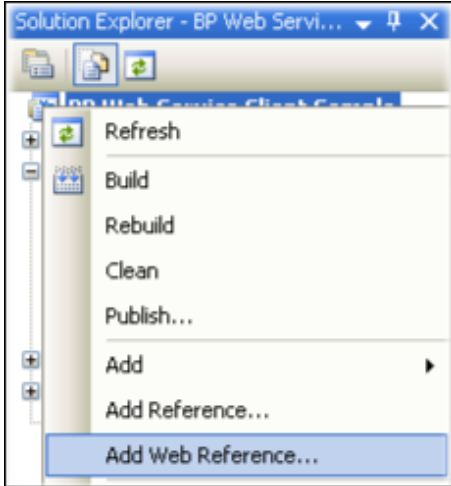


3. System.Web.Services 를 선택하고 확인을 클릭합니다.

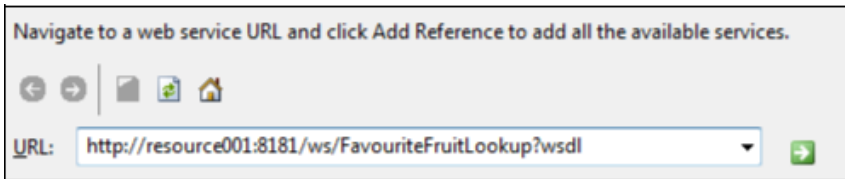


5.4.5. Add Web References to each of the WSDLs

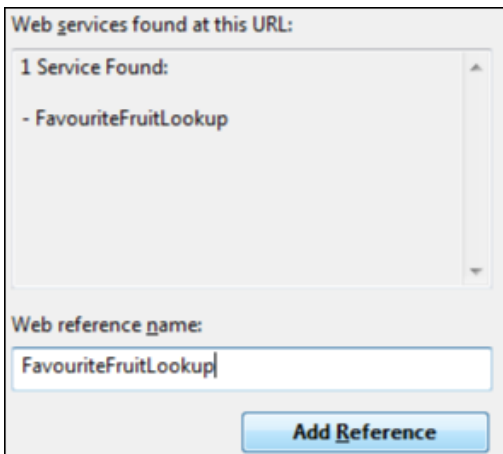
1. solution explorer 에서 프로젝트 이름을 마우스 오른쪽 단추로 클릭하고 Add Web Reference 를 선택하십시오. (최신 버전의 Visual Studio 를 사용하는 경우, Add Service Reference 를 선택하고, Advanced 를 클릭하고, Add Web Reference 를 클릭하십시오.)



2. FavouriteFruitLookup 에 대한 WSDL 의 URL 을 입력하십시오.



3. 그러면 Visual Studio 가 WSDL 문서를 검사합니다.
Web Reference 의 이름을 FavouriteFruitLookup 으로 설정하십시오.



4. NumericOperations WSDL 에 대해 위의 단계를 반복하고 이름을 NumericOperations 로 지정하십시오.

5.4.6. Example Code: Interact with a Business Object based Web Service

Business Objects 는 많은 개별 작업으로 구성될 수 있으며 이들 각각은 노출된 웹 서비스를 통해 사용할 수 있습니다. 예를 들어 Business Object **NumericOperations** 에는 **Add Numbers** 및 **Subtract Numbers** 의 두 가지 액션이 있습니다.

Business Object 기반 웹 서비스는 사용하기 전에 초기화해야 합니다. 다음 두 가지 방법을 사용할 수 있습니다.

- **Manual Initialisation:** 비즈니스 객체의 수명주기를 완전히 제어할 수 있습니다. **Intialise** 작업을 사용하여 세션 ID 를 받은 다음 후속 작업에서 **bplInstance** 입력 매개 변수를 채우는 데 사용됩니다.
- **Auto-Initialisation:** Blue Prism 이 사용자를 대신하여 Business Object 의 초기화를 처리하므로 Business Object 작업을 호출하는 데 필요한 단계 수를 줄입니다. 매개 변수 **bplInstance** 를 “auto”값으로 설정하여 수행됩니다.

사용할 초기화 모드를 선택할 때 다음 사항을 고려해야 합니다.

- 자동 초기화를 사용하면 Blue Prism Business Object 내에 정의된 모든 전역 변수가 각 관련 웹 서비스 요청에서 초기화된 상태로 유지됩니다.
- 수동 초기화를 사용하는 경우 Business Object 가 초기화될 때마다 Control Room 내에 세션이 표시됩니다. 자동 초기화와 마찬가지로 특정 웹 서비스에 대해 수행되는 모든 작업에 단일 세션이 사용됩니다.

초기화 방법 선택에 관계없이 필요한 단계는 다음과 같습니다.

1. **TestNumericWebService** 라는 하위 프로시저를 만듭니다.

```
Private Sub TestNumericWebService()  
End Sub
```

2. 애플리케이션이 실행될 때 콘솔에 정보를 표시하고 웹 서비스와 상호 작용하기 위한 프록시 클래스의 새 인스턴스를 작성하려면 다음 행을 삽입하십시오.

```
Console.WriteLine("Numeric web Service Test")  
Dim NOS As New NumericOperations.NumericOperationsService
```

3. Blue Prism 으로 인증을 처리하는 줄을 다음과 같이 추가합니다.

```
NOS.Credentials = New System.Net.NetworkCredential("admin", "admin")
```

사용자 이름과 암호는 로그인 권한이 있는 사용자의 Blue Prism 자격 증명에 유효해야 하며 Control Room 에서 프로세스를 실행해야 합니다. 사용자는 또한 Business Object 를 실행하기 위한 접근 권한이 필요합니다.

4. **수동 초기화**를 사용하는 경우 초기화를 수행하고 세션 ID 를 반환하는 다음 줄을 추가합니다. 이 세션 ID 는 Blue Prism 이 동일한 Business Object 에서 동시에 실행되는 여러 인스턴스를 구별할 수 있도록 Business Object 작업 방법 중 하나가 요청될 때 제공됩니다.

```
Dim SessionID As String = NOS.Initialise()  
Console.WriteLine("Initialised")
```

자동 초기화를 사용하는 경우 대신 다음 줄을 추가하여 사용하면 Blue Prism 에서 자동으로 초기화를 처리하도록 합니다.

```
Dim SessionID As String = "auto" 'set string value as auto
```

5. 두 가지 다른 작업을 호출하고 결과를 출력하는 다음 코드를 포함합니다. 각각은 초기화 작업에 의해 반환된 세션 ID 를 전달하거나 자동 초기화를 사용하는 경우 "auto"라는 텍스트를 전달합니다.

```
'Call the action method "Add Numbers"
Dim Result1 As Decimal = NOS.AddNumbers(SessionID, 3, 5)
Console.WriteLine("Added 3 and 5 to get " & Result1.ToString)

'Call the action method "Subtract Numbers"
Dim Result2 As Decimal = NOS.SubtractNumbers(SessionID, 3, 5)
Console.WriteLine("Subtracted 5 from 3 to get " & Result2.ToString)
```

6. 마지막으로 **수동 초기화**를 사용하는 경우 세션을 닫는 정리 작업을 수행해야 합니다. 이를 통해 Blue Prism 은 세션을 닫고 관련 리소스를 폐기할 수 있습니다.

```
Console.WriteLine("Cleaning up ...")
NOS.Cleanup(SessionID)
```

자동 초기화를 사용하는 경우 정리 작업이 필요하지 않습니다.

전체 절차는 다음과 같이 나타납니다.

- **Manual Initialisation (수동 초기화)**

```
Private Sub TestNumericWebService()
    Console.WriteLine("Numeric Web Service Test")
    Dim NOS As New NumericOperations.NumericOperationsService

    NOS.Credentials = New System.Net.NetworkCredential("admin", "admin")
    Dim SessionID As String = NOS.Initialise()
    Console.WriteLine("Initialised")

    'Call the action method "Add Numbers"
    Dim Result1 As Decimal = NOS.AddNumbers(SessionID, 3, 5)
    Console.WriteLine("Added 3 and 5 to get " & Result1.ToString)

    'Call the action method "Subtract Numbers"
    Dim Result2 As Decimal = NOS.SubtractNumbers(SessionID, 3, 5)
    Console.WriteLine("Subtracted 5 from 3 to get " & Result2.ToString)

    Console.WriteLine("Cleaning up ...")
    NOS.Cleanup(SessionID)
End Sub
```

- **Auto-Initialisation (자동 초기화)**

```
Private Sub TestNumericWebService()
    Console.WriteLine("Numeric Web Service Test")
    Dim NOS As New NumericOperations.NumericOperationsService

    NOS.Credentials = New System.Net.NetworkCredential("admin", "admin")
    Dim SessionID As String = "auto"
    'Console.WriteLine("Initialised")

    'Call the action method "Add Numbers"
    Dim Result1 As Decimal = NOS.AddNumbers(SessionID, 3, 5)
    Console.WriteLine("Added 3 and 5 to get " & Result1.ToString)

    'Call the action method "Subtract Numbers"
    Dim Result2 As Decimal = NOS.SubtractNumbers(SessionID, 3, 5)
    Console.WriteLine("Subtracted 5 from 3 to get " & Result2.ToString)

End Sub
```

5.4.7. Example Code: Interact with a Process based Web Service

Business Objects 와 달리 프로세스는 각 인스턴스에 대해 한 번씩 실행되며 진입점이 하나 뿐이며 명시적인 초기화 또는 정리 작업이 필요하지 않습니다.

1. TestFruitLookupWebService()라는 하위 프로시저를 만듭니다.

```
Private Sub TestFruitLookupWebService()
End Sub
```

2. 다음 줄을 사용해서 응용 프로그램이 실행될 때 콘솔에 정보를 표시하고, 웹 서비스와 상호 작용하기 위한 프록시 클래스의 새 인스턴스를 만드십시오.

```
Console.WriteLine("FruitLookup Web Service Test")
Dim FFL As New FavouriteFruitLookup.FavouriteFruitLookupService
```

3. Blue Prism 으로 인증을 처리하는 다음 줄을 추가합니다.

```
FFL.Credentials = New System.Net.NetworkCredential("admin", "admin")
```

사용자 이름과 암호는 로그인하고 Control Room 에서 프로세스를 실행할 권한이 있는 사용자의 Blue Prism 자격 증명으로 유효해야 합니다.

4. 여러 이름을 갖는 문자열 배열을 정의하고 이어서 각각의 이름에 대해서 반복해서 웹 서비스를 쿼리하여, 가장 좋아하는 과일이 무엇인지 알아내는 코드를 추가하십시오.

```
For Each Person As String In New String() _
{"Bashful", "Doc", "Dopey", "Grumpy", "Happy", "Sleepy", "Sneezy"} Dim
    Fruit As String = FFL.FavouriteFruitLookup(Person)
    Console.WriteLine(String.Format("{0}'s favourite fruit is {1}", Person, Fruit))
Next
```

전체 절차는 다음과 같이 나타납니다.

```
Private Sub TestFruitLookupWebService()
    Console.WriteLine("FruitLookup Web Service Test")
    Dim FFL As New FavouriteFruitLookup.FavouriteFruitLookupService
    FFL.Credentials = New System.Net.NetworkCredential("admin", "admin")
    For Each Person As String In New String() _
        {"Bashful", "Doc", "Dopey", "Grumpy", "Happy", "Sleepy", "Sneezy"}
        Dim Fruit As String = FFL.FavouriteFruitLookup(Person)
        Console.WriteLine(String.Format("{0}'s favourite fruit is {1}", Person, Fruit))
    Next

    Console.WriteLine("End of FruitLookup Test")
End Sub
```

5.4.8. Example Code: Trigger the Interactions when the application is run

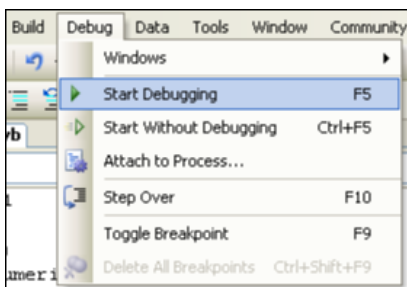
이 예제는 콘솔 애플리케이션 내의 기본 프로시저를 수정하여 Business Object 기반 웹 서비스 및 프로세스 기반 웹 서비스와의 위와 같이 상호 작용하도록 트리거합니다.

1. 하위 프로시저를 호출하도록 **main** 프로시저를 수정하십시오.

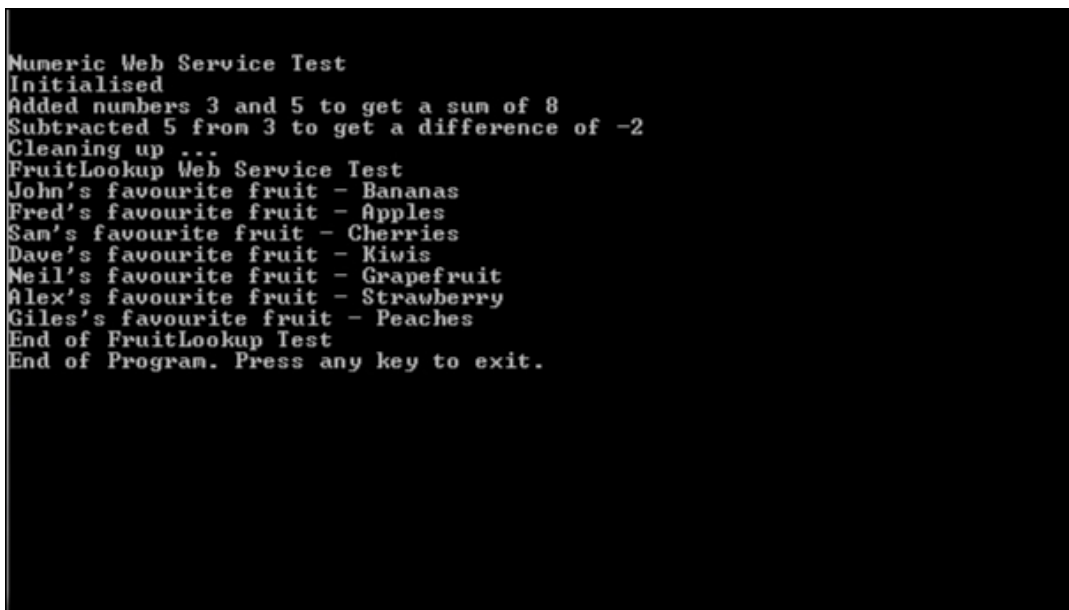
```
Sub Main()
    TestNumericWebService()
    TestFruitLookupWebService()

    Console.WriteLine("End of Program. Press any key to exit.")
    Console.ReadKey()
End Sub
```

2. Visual Studio 에 프로젝트를 저장하고 디버깅 도구를 사용하여 실행합니다.



3. 콘솔 애플리케이션 출력은 아래 스크린 샷과 유사해야 합니다.



6. Frequently Asked Questions

1. 어떤 버전의 Blue Prism 에서 Business Object 및 프로세스를 웹 서비스로 노출하기 위한 마법사가 처음으로 포함되었습니까?

Blue Prism Business Object 및 Blue Prism 프로세스를 웹 서비스로 노출하는 마법사 기능은 버전 v4.2.35 에서 처음으로 도입되었습니다.

2. Blue Prism 에서 사용하는 웹 서비스의 시간 제한 값은 얼마입니까?

3.0.84 버전 이전에는 타사 웹 서비스가 100 초의 하드 코딩된 제한 시간 값을 갖도록 구성되었습니다. Blue Prism 의 후속 버전에서는 각 웹 서비스에 대해 이 기본값을 재구성할 수 있습니다.

3. Blue Prism 은 바이너리 웹 서비스 데이터를 처리할 수 있습니까?

3.0.76 버전에 도입된 Blue Prism 은 노출된 웹 서비스와 소비된 웹 서비스 모두에 대한 입력 및 출력으로 바이너리 데이터를 지원합니다. 이것은 파일(문서, 이미지, 그림 등)을 입력 또는 출력 매개 변수로 전달하는 기능을 제공합니다.

4. Blue Prism 은 HTTP 를 통해 제공되는 타사 웹 서비스를 사용할 수 있습니까?

예, 그러나 그러한 웹 서비스를 사용할 Blue Prism 런타임 리소스가 웹 서비스를 보호하는 데 사용되는 인증서를 신뢰하도록 구성하는 것이 중요합니다. 여기에는 공용 인증 기관이 인증서를 발급하는지 확인하거나 인증서가 각 런타임 리소스의 적절한 영역에 추가되었는지 확인하는 것이 포함될 수 있습니다.

5. Blue Prism 은 RESTful 웹 서비스를 사용할 수 있습니까?

주로 다양한 SOAP 웹 서비스에 대한 지원이 제공되지만 REST 웹 서비스에 대한 지원은 REST, HTTP, JSON 및 Oauth 와 같은 일련의 예제 VBO 를 사용하여 적절한 프로그래밍 경험이 있는 개발자에게 제공됩니다.

6. Blue Prism 은 구조화된 웹 서비스 데이터를 처리할 수 있습니까?

버전 3.5.33 에 도입된 Blue Prism 은 중첩된 컬렉션을 지원합니다. 이를 통해 Blue Prism 은 웹 서비스를 노출하거나 사용할 때 구조화된 데이터를 보내고 받을 수 있습니다.

7. Blue Prism 웹 서비스에 액세스하려면 어떤 인증이 필요합니까?

Blue Prism 웹 서비스는 HTTP 인증을 사용하여 보호됩니다. Blue Prism 사용자 계정은 노출된 Blue Prism 웹 서비스를 사용할 각 제 3 자 공급자에 대해 구성하는 것이 좋습니다. 그런 다음 이러한 사용자의 자격 증명을 HTTP 헤더 인증 프로세스의 일부로 사용할 수 있습니다.

8. Blue Prism 이 만들 수 있는 웹 서비스 요청의 크기에 제한이 있습니까?

Blue Prism 은 요청할 수 있는 크기에 제한이 없습니다. 그러나 웹 서비스 서버는 일반적으로 제한(일반적으로 4MB)을 부과합니다.

9. 재귀적 정의를 포함하는 WSDL 사용에 대한 지원이 제공됩니까?

WSDL 내의 재귀 정의는 Blue Prism 에서 지원되지 않습니다. DataCategory 에 대한 재귀적 정의를 포함하는 WSDL 의 예제 스키마는 다음과 같습니다.

```
<complexType name="DataCategory">
  <sequence>
    <element name="childCategories" type="tns:DataCategory" minOccurs="0"
maxOccurs="unbounded"/>
    <element name="label" type="xsd:string"/>
    <element name="name" type="xsd:string"/>
  </sequence>
</complexType>
```

7. Support

이 문서를 이용하면서 추가 지원이 필요한 경우, Blue Prism 계정 관리자 또는 기술 지원(support@blueprism.com)에 문의하십시오.