

IBM Intelligent Operations Center 1.6 Programming Guide

Unleash the full potential of the programming model

Extend and customize the base solution

Learn from practical examples and sample code



Hisham Abdel-Hafez
Sriram Balakrishnan
Jamie Caffrey
Eduardo Francellino
Sunil Mishra
Teresa Nascimento
Jayashree Ravichandran
Craig Scott
Nikolay Vlasov



International Technical Support Organization

**IBM Intelligent Operations Center 1.6 Programming
Guide**

May 2014

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (May 2014)

This edition applies to Version 1, Release 6 Modification 01 of IBM Intelligent Operations Center (5725-D69).

© Copyright International Business Machines Corporation 2014.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
Authors	xi
Now you can become a published author, too!	xiv
Comments welcome	xv
Stay connected to IBM Redbooks	xv
Chapter 1. IBM Intelligent Operations Center programming model overview	1
1.1 IBM Intelligent Operations Center components	2
1.2 IBM Intelligent Operations Center services API	3
1.2.1 IBM Intelligent Operations Center services API design	5
1.3 REST services	6
1.3.1 Resource URIs	6
1.3.2 REST services access	8
1.3.3 Authentication	8
1.3.4 Client-side service invocation sample	9
1.3.5 General guidelines	12
1.3.6 Http response status codes	13
1.4 RestStore	13
1.4.1 Purpose	13
1.4.2 Benefits	14
1.4.3 Design	14
1.4.4 Using RestStore	18
1.5 Java APIs	18
1.5.1 Implementation details	18
1.5.2 Server-side service invocation sample	19
1.6 Session persistence	25
1.7 List of IBM Intelligent Operations Center services	27
1.8 Extension points	32
1.9 Programming considerations for high availability	34
Chapter 2. Sample scenario	37
2.1 Sample scenario overview	38
2.2 Planning for the event	39
2.3 During the event	40
2.3.1 Keeping track of crowd sizes	40
2.3.2 Keeping track of first aid incidents	41
2.3.3 Responding to incidents	42
2.4 After the event	42
Chapter 3. Integrating with data sources	45
3.1 The data import process	46
3.1.1 Data import process overview	46
3.2 Configuring data sources	49
3.2.1 Key considerations when planning to import data into IBM Intelligent Operations Center	50
3.2.2 Use case overview	52

3.2.3 Database table data source	54
3.2.4 CSV data source	55
3.2.5 CAP data source	56
3.2.6 User interface entry data source	58
3.3 Using APIs to send data into the system	58
3.3.1 API overview	58
3.3.2 Sending data items using the API	59
3.4 Extension points	62
3.4.1 External service URL data sources	63
3.4.2 Extension properties in the resource definition	63
3.5 Troubleshooting data sources	64
3.5.1 Data items do not reach data source target table in the system database	64
3.5.2 Data stops to be imported after a database failure	66
3.5.3 Things to check if data items are not displayed on the map	66
Chapter 4. Tracking key performance indicators	69
4.1 Overview	70
4.1.1 Terminology	70
4.1.2 IBM Intelligent Operations Center capabilities	71
4.1.3 Types of KPIs	72
4.1.4 Tooling choices	72
4.2 Use case overview	73
4.2.1 Actors	73
4.2.2 Desired outcome	73
4.2.3 IBM Intelligent Operations Center programming artifacts	73
4.3 Setting up the development environment	74
4.4 Implementation summary	74
4.5 Implementation details	74
4.5.1 Identifying input data	74
4.5.2 Defining the monitor model	75
4.5.3 Creating the KPI	88
4.5.4 Establishing KPI relationships	91
4.5.5 Visualizing the new KPI	93
4.6 Problem determination	94
4.6.1 IBM Intelligent Operations Center configuration	94
4.6.2 IBM Business Monitor configuration	96
4.6.3 IBM Business Monitor credentials	97
4.6.4 IBM Business Monitor SSL certificates	98
4.6.5 Message routing	101
4.7 Programmatic access to KPIs	102
4.8 Additional considerations	105
4.8.1 Tips and good practices	106
4.8.2 Performance considerations	106
Chapter 5. Driving standard operating procedures	109
5.1 Managing SOPs in IBM Intelligent Operations Center	110
5.1.1 Roles and permissions	111
5.1.2 Configuring SOP definitions	115
5.1.3 Editing definitions	116
5.1.4 Working with drafts and submitting for approval	117
5.1.5 Approving or disapproving drafts	118
5.1.6 Launching SOPs	119
5.1.7 Viewing SOP instances	122

5.1.8	Reverting to a specific version	124
5.1.9	Exporting and importing SOPs	124
5.2	Configuring activities for different scenarios	125
5.2.1	Configuring activities to execute sequentially	127
5.2.2	Configuring required activities	129
5.2.3	Configuring manual activities	130
5.2.4	Configuring automation activities	132
5.2.5	Configuring If-Then-Else activities	134
5.2.6	Configuring notification activities	136
5.2.7	Configuring REST service activities	141
5.2.8	Configuring SOP activities	143
5.3	SOP REST service	144
5.3.1	Retrieving all SOP definitions	144
5.3.2	Retrieving an SOP definition	147
5.3.3	Retrieving activity definitions	147
5.3.4	Working with SOP instances	148
5.3.5	Working with activity instances	150
5.3.6	Using REST services to analyze the historical executions of an SOP	153
5.4	Use case overview	154
5.4.1	Actors	155
5.4.2	Desired outcome	155
5.4.3	IBM Intelligent Operations Center programming artifacts	155
5.5	Manage event SOP	156
5.5.1	Implementation summary	156
5.5.2	Implementation details	156
5.5.3	Querying SOP instance data	171
5.6	Manage crowd incident SOP	177
5.6.1	Implementation summary	178
5.6.2	Implementation details	178
	Chapter 6. Customizing the operations display	191
6.1	User interface customization overview	192
6.1.1	Content panel customization	192
6.1.2	Creating custom data source actions	192
6.1.3	Adding custom filter-driven results charts	193
6.1.4	Customizing preview cards	194
6.1.5	Creating custom portlets	195
6.1.6	Adding custom help links	195
6.1.7	Customizing the More Actions menu	196
6.1.8	Creating custom portal pages	196
6.2	Customizing the content panel	197
6.2.1	Implementation summary	197
6.2.2	Setting up the development environment	198
6.2.3	Implementation details	198
6.3	Creating custom data source actions	219
6.3.1	Implementation summary	219
6.3.2	Setting up the development environment	220
6.3.3	Implementation details	220
6.3.4	Visualizing the results	225
6.4	Adding filter-driven result charts	227
6.4.1	Custom filter-driven result chart creation example	229
6.5	Customizing preview cards	233
6.5.1	Implementation summary	234

6.5.2	Setting up the development environment	234
6.5.3	Implementation details	234
6.6	Creating custom portlets	239
6.6.1	Custom portlet development environment	239
6.6.2	Deploying custom portlets.	242
6.6.3	Using RestStore	246
6.6.4	Using the translation widget and internationalization service	248
6.7	Adding custom help links	252
6.7.1	Adding an Eclipse plug-in with custom help articles	252
6.7.2	Creating a new Help menu item and adding it to the Help menu	252
6.8	Customizing the More Actions menu	255
6.9	Creating custom solution pages	257
6.9.1	IBM Intelligent Operations Center solution themes.	257
6.9.2	Creating custom portal pages	259
6.9.3	Change the city name on the top of the user interface	265
6.9.4	Customizing titles and IBM logos in IBM Intelligent Operations Center	268
Chapter 7. Visualizing information with maps		269
7.1	GIS terminology	270
7.1.1	Generic GIS terminology.	270
7.1.2	GIS terminology specific to IBM Intelligent Operations Center	271
7.2	Creating maps.	274
7.2.1	Creating a base map.	274
7.2.2	Creating location maps	275
7.2.3	Configuring map settings	282
7.2.4	Map extension points	283
7.3	Use case overview	285
7.3.1	Actors	285
7.3.2	Desired outcome	285
7.3.3	IBM Intelligent Operations Center programming artifacts	285
7.4	Implementation summary	286
7.5	Implementation details	286
7.5.1	Adding data sources	286
7.5.2	Creating emergency events	287
7.5.3	Finding nearby medical units	290
Chapter 8. Developing mobile applications		299
8.1	Technology overview	300
8.1.1	IBM Intelligent Operations Center REST services	300
8.1.2	IBM Worklight	301
8.2	Use case overview	302
8.2.1	Actors	302
8.2.2	Desired outcome	302
8.2.3	IBM Intelligent Operations Center programming artifacts	302
8.3	Implementation summary	302
8.3.1	Setting up the development environment	302
8.4	Implementation details	303
8.4.1	Creating a new IBM Worklight project.	303
8.4.2	Building the IBM Worklight adapters.	305
8.5	Deploying the artifacts.	319
8.5.1	Building and deploying mobile devices	319
8.5.2	Deploying server-side components.	319
8.6	Visualizing the results	320

Chapter 9. Integrating with external systems	321
9.1 Overview	322
9.2 Integration with IBM SmartCloud Control Desk through SOP automation activity . . .	322
9.2.1 Actors	322
9.2.2 Desired outcome	322
9.2.3 IBM Intelligent Operations Center programming artifacts	323
9.2.4 Implementation summary	323
9.2.5 Implementation details	323
9.2.6 Verify IBM SmartCloud Control Desk settings for web service access	326
9.2.7 Create the SOP to trigger the job plan through automation activity	327
9.3 Integration through REST service activity	328
9.4 Routing data to the integration topic	331
9.4.1 Desired outcome	332
9.4.2 IBM Intelligent Operations Center programming artifacts	332
9.4.3 Implementation summary	332
9.4.4 Implementation details	332
9.5 Using an external service URL	335
9.5.1 Custom service requirements	336
9.5.2 Adding an external service URL	341
9.6 Other options for integration	348
Chapter 10. Creating and viewing reports	349
10.1 Cognos reports versus Dojo charts	350
10.2 Use case overview	350
10.2.1 Actors	350
10.2.2 Desired outcome	351
10.2.3 IBM Intelligent Operations Center programming artifacts	351
10.3 Setting up the development environment	351
10.4 Implementation summary	352
10.5 Implementation details	353
10.5.1 Verify the prerequisite configuration	353
10.5.2 Determine the column names from the data source	354
10.5.3 Create the simple report	355
10.5.4 Create a report across multiple data sources	359
10.6 Deploying Cognos reports	371
10.7 Displaying Cognos reports	371
10.8 Problem determination	372
10.9 Additional considerations	373
Appendix A. Development tools	375
Development activities and tools	376
IBM Business Monitor and IBM Rational Application Developer special note	378
IBM Worklight Consumer Edition special note	378
IBM Cognos Framework Manager and IBM Cognos Report Studio special note	379
Setting up IBM Rational Application Developer for WebSphere Software 8.5	379
Installing and configuring IBM Cognos Framework Manager	380
Appendix B. Samples provided with this book	383
Extracting the samples compressed file	385
Integrating with data sources samples	386
Integrating with external systems samples	387
Developing mobile application samples	387
Importing the sample projects into IBM Worklight Studio workspace	388
IBM Intelligent Operations Center programming model overview samples	389

Customizing the operations display samples	389
Key considerations for using Rational Application Developer	391
Importing the sample projects into IBM Rational Application Developer workspace	392
Installing IBM Intelligent Operations Center JARS in the development environment	395
Appendix C. Additional material	397
Locating the web material	397
Using the web material.	397
System requirements for downloading the web material	397
Downloading and extracting the web material	398
Related publications	399
IBM Redbooks	399
Online resources	399
Help from IBM	400

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.

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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Cognos®	ILOG®	Smarter Cities®
CPLEX®	Lotus®	SPSS®
DB2®	Rational®	Tivoli®
Global Business Services®	Redbooks®	WebSphere®
IBM®	Redbooks (logo)  ®	Worklight®
IBM SmartCloud®	Sametime®	

The following terms are trademarks of other companies:

Worklight is trademark or registered trademark of Worklight, an IBM Company.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

Preface

IBM® Intelligent Operations Center is an integrated solution. It provides a rich set of capabilities and line of business tools that business users with domain expertise and no technical background can use without customization.

IBM Intelligent Operations Center also provides services and extension points that developers can use to extend the IBM Intelligent Operations Center standard functions and develop capabilities specific to the domain and client requirements.

IBM Intelligent Operations Center includes an application-based programming model that supports all the interactions with the solution components. The programming model is based on industry standard Representational State Transfer (REST) and Java technologies. IBM Intelligent Operations Center includes a full set of REST and Java application programming interfaces (APIs) that provide a simplified development environment and make the platform easy to extend and customize for a large community of developers.

This IBM Redbooks® publication gives a broad understanding of the IBM Intelligent Operations Center 1.6.0.1 programming model and available extension points. Many of the chapters describe working examples and usage scenarios that demonstrate how to extend the IBM Intelligent Operations Center base platform. This book includes sample code that can be downloaded from the IBM Redbooks website.

The target audience for this book consists of solution architects, developers, technical consultants, and solution administrators who will learn the following information:

- ▶ The options available to extend the IBM Intelligent Operations Center solution programmatically
- ▶ How to configure customizations tailored to specific client requirements
- ▶ How to use the available configuration tools to configure the solution without requiring programming

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.



Hisham Abdel-Hafez is a certified expert IT Specialist in Cairo Technology Development Center (CTDC) in IBM Egypt. He has more than seven years of experience in application development and technical team leadership. Hisham excels in the area of IBM WebSphere Portal and IBM WebSphere® Application server solutions. Hisham's current focus is IBM Worklight® and related IBM MobileFirst technologies. His experience with mobile application development started in the Symbian days and progressed to reach the IBM Worklight era. Hisham has 11 technical certifications, including IBM Worklight mobile application development.



Sriram Balakrishnan is an Executive Architect with the IBM Sales & Distribution division where he provides architectural direction for the Smarter Lifecycle Asset Management (SLAM) group. He has been with IBM since 2002 and has more than 20 years of experience in software development and architecture. He holds a Master of Technology in Software Engineering from the National University of Singapore.



Jamie Caffrey is a Client Solution Professional for IBM Smarter Cities® in IBM Software Group in the UK. Jamie works with both public and private sector clients, helping them develop new solutions and insights out of the wealth of information locked within the many silos that exist in their organizations. Jamie focuses on Semantic Web, Open Data, and Contextual Computing in city and business environments. Jamie has worked for IBM for 13 years; his previous assignment was in IBM Global Business Services® consulting. Jamie has a degree in Information Technology from Edinburgh Napier University and in Engineering from Heriot-Watt University.



Eduardo Francellino is a Client Solution Professional in IBM Software Group, IBM Brazil. Eduardo is a subject matter expert (SME) on IBM Smarter Cities in Latin America. Eduardo's focus is on IBM Intelligent Operations Center, IBM Intelligent Transportation, and IBM Intelligent Water. Before joining IBM, Eduardo worked for several companies as a software developer in various industry sectors, such as oil and gas, telecommunications, and financial markets. He holds a degree in Information Technology from Estácio de Sá University and an MBA in Software Engineering from Federal University of Rio de Janeiro. He is the co-author of the IBM Redbooks publication, *IBM Intelligent Operations Center for Smarter Cities Administration Guide*, SG24-8061.



Sunil Mishra is a Lead Developer for IBM Intelligent Operation Center, in the IBM Research Triangle Park (RTP) laboratory in Raleigh, North Carolina. Sunil has been part of the IBM Intelligent Operation Center development team since version 1.0. Sunil's previous assignments include development of several IBM products, such as IBM Premises RFID server, IBM WebSphere Sensor Event, and IBM Real-Time Asset Locator. Sunil has more than 17 years of experience in the IT industry. He has filed several patents with the United States Patent and Trademark Office (USPTO). Sunil holds a Bachelor's Degree in Business from Mumbai University, India.



Teresa Nascimento is an IBM Industry Solutions Architect with 12 years experience in the IT industry. Teresa is the technical leader for IBM Smarter Cities and Government in IBM Brazil. Teresa focuses on public sector clients in Brazil. Her primary areas of expertise include city management, public safety, transportation, and water. Teresa's main responsibility is to lead the design and development of cross-brand solutions and support the delivery of projects that address critical requirements for cities and government entities.



Jayashree Ravichandran is an IBM Certified IT Architect in IBM Industry Solutions and IBM Smarter Cities, IBM US. Jayashree helps clients to identify requirements, design, and architect solutions based on the IBM Smarter Cities portfolio. She also acts as a bridge between the client's business requirements and IBM development teams helping to build the solutions that best fit the client's needs. Her previous assignments include software development for IBM Real-Time Asset Locator and IBM Integrated Information Core. She has over 15 years of experience in the software industry, in all phases of the software development lifecycle. Her areas of expertise include solution architecture, design, software development, and deployment planning. She holds a degree in Electronics and Communication Engineering.



Craig Scott is a Software Support Specialist in IBM Australia. He has 24 years of experience in the IT industry and over fourteen years of experience with IBM products. Since 2010, he has been working for IBM Industry Solutions Support assisting clients and IBM employees with IBM Intelligent Information Center, IBM Intelligent Transportation, IBM Intelligent Water, IBM City Planning and Optimization, and the IBM Defense Operations Platform deployments. Craig holds a degree in Computer Science from the University of Canberra. His areas of expertise include IBM WebSphere Application Server, IBM WebSphere Portal, IBM Tivoli Access Manager, IBM Tivoli® Monitoring, IBM DB2®, IBM Content Manager, and Linux.



Nikolay Vlasov is a Technical Sales Specialist in IBM Singapore. Nikolay helps clients in the ASEAN region to identify their business requirements and design solutions based on IBM Intelligent Operations Center and other products in the IBM Smarter Cities portfolio. Nikolay's previous assignments include technical sales for IBM WebSphere Portal and IBM Collaboration solutions in Russia and CIS countries. Nikolay holds a Master's Degree in Computer Science from Moscow State University, Russia.

The project that produced this publication was managed by **Marcela Adan**, IBM Redbooks Project Leader - International Technical Support Organization, Raleigh Center.

Thanks to the following people for their contributions to this project:

Kai Arrowood
Ron Baker
Aidan Butler
James Condon
Steven De Gennaro
Chris Durso
Ken Greenle
Shauna Harris
Paul Holthues
Garret O’Keeffe
Bob Patten
Dave Rusnak
Allen Smith
Dave Soroka
IBM Industry Solutions Development

Gareth Burke
Vanadis Crawford
Annette Sohor
IBM Industry Solutions Information Development

Tony Carrato
IBM Industry Solutions Software, Architecture, and Technical Strategy, IBM US

Adam Holley
Smarter Cities Advanced Technology and Assets

Robin James
IBM Industry Solutions Lab Services

Pam Nesbitt
IBM Smarter Cities Client Success Consulting Services, IBM US

Diwakar Puthalapat
IBM Smarter Cities, Industry Solutions, IBM India

Jun Xia Zhou
IBM Smarter Cities Cloud Delivery, IBM China

Deana Coble
Richard Conway
Shawn Tooley
IBM International Technical Support Organization, Raleigh Center

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:
ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



IBM Intelligent Operations Center programming model overview

This chapter describes the IBM Intelligent Operations Center Programming Model, which provides services that can be used to access the IBM Intelligent Operations Center data and extend the capabilities of the solution programmatically.

This chapter describes the design and use of the services from client-side code via the Representational State Transfer (REST) service interface and from server-side code via the Java APIs. This chapter provides an example that prints the list of installed products and their versions, implemented in both ways.

This chapter includes the following topics:

- ▶ 1.1, “IBM Intelligent Operations Center components” on page 2
- ▶ 1.2, “IBM Intelligent Operations Center services API” on page 3
- ▶ 1.3, “REST services” on page 6
- ▶ 1.4, “RestStore” on page 13
- ▶ 1.5, “Java APIs” on page 18
- ▶ 1.6, “Session persistence” on page 25
- ▶ 1.7, “List of IBM Intelligent Operations Center services” on page 27
- ▶ 1.8, “Extension points” on page 32
- ▶ 1.9, “Programming considerations for high availability” on page 34

Note: For information about the sample code for the examples described in this chapter, see Appendix B, “Samples provided with this book” on page 383.

1.1 IBM Intelligent Operations Center components

At a high level, the structure of IBM Intelligent Operations Center can be divided into major components, subsystems, and services. Figure 1-1 shows a high-level view of IBM Intelligent Operations Center with the servers and main components running on each server.

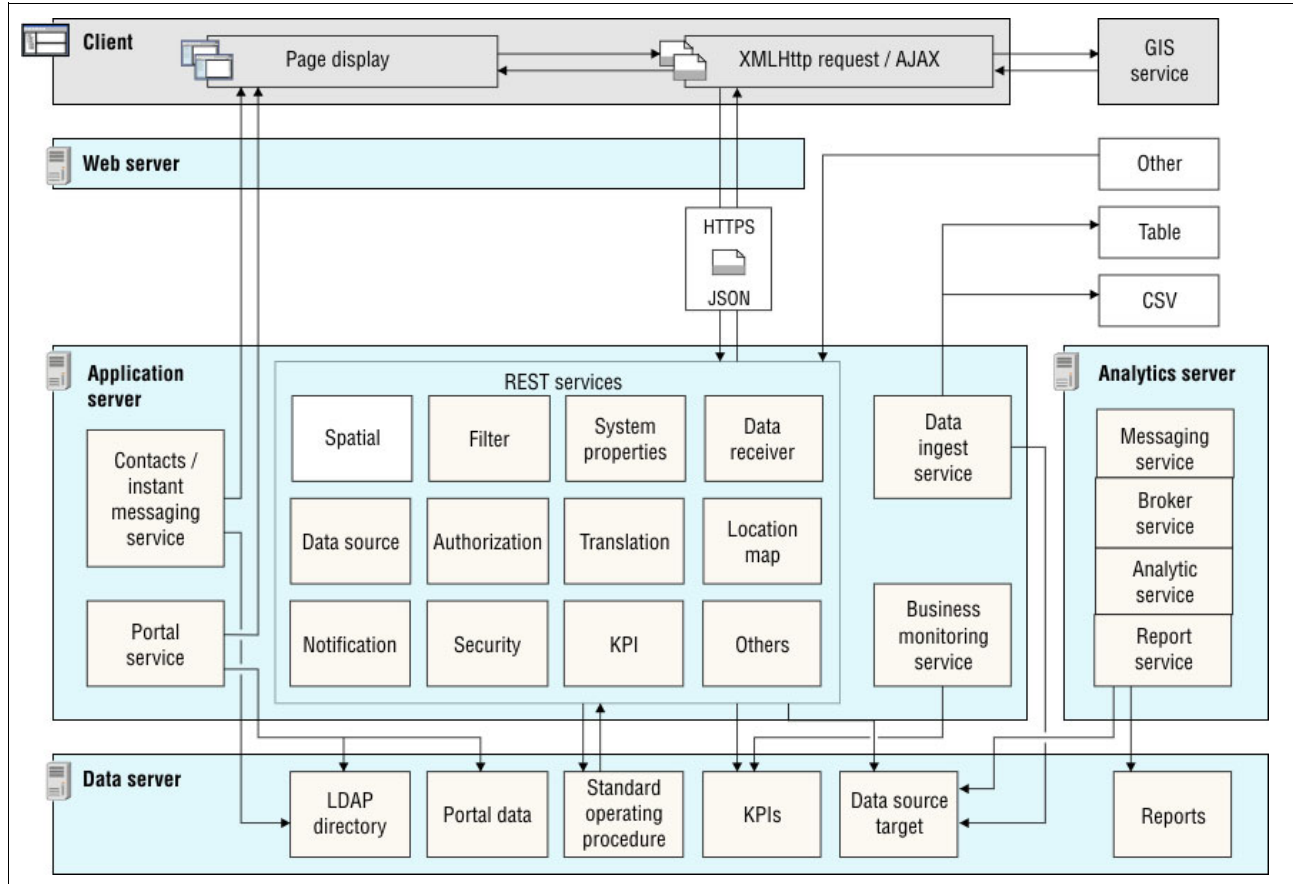


Figure 1-1 IBM Intelligent Operations Center components

The following components in Figure 1-1 are described:

- ▶ Client

IBM Intelligent Operations Center provides web-based, one-stop portals to event information, overall status, and details. The user interface (UI) presents customized information in various preconfigured views in common formats. All information is displayed through easy-to-use dashboards.

- ▶ Web server

The web server manages client requests to IBM Intelligent Operations Center.

- ▶ Application server

The application server provides a set of services for accessing and visualizing data that is managed by IBM Intelligent Operations Center. The data ingest service is used to import data from disparate external sources, such as databases and files. Because of the variety of data that can be supplied from these external sources, the data is normalized into a common format that can be used to calculate key performance indicators, trigger standard operating procedures and notifications, and provide spatial-temporal data.

The contacts and instant messaging service provides capabilities that enable effective, real-time communication.

The business monitoring service monitors incoming data records, and uses the information that is contained in the data records to generate key performance indicators. In the user interface layer that is provided by the Portal service, users can view key performance indicators, standard operating procedures, notifications, and reports. Users can also view spatial-temporal data on a geospatial map, or they can view specific details that represent a building or a stadium either on a location map, or in a list view.

The application server provides security services that ensure that only authorized users and groups can access data.

- ▶ Data server

The data server stores all the databases that are used in IBM Intelligent Operations Center.

- ▶ Analytics server

The analytics server contains the following components that might be used by either custom solutions, or by other domain applications:

- Messaging service and broker service

The messaging service that is provided by WebSphere MQ, and the broker service that is provided by WebSphere Message Broker, can be used to receive and transform data in formats that IBM Intelligent Operations Center does not support. When the data is transformed, it can be written either to a database table or to a comma-separated value (CSV) file that IBM Intelligent Operations Center can process.

- Analytic service

The analytic service contains IBM SPSS® WebSphere Business Modeler and IBM ILOG® CPLEX® Optimization Studio.

- Report service

You can use the report service that is provided by IBM Cognos® Report Studio to generate reports.

For more information about IBM Intelligent Operations Center high-level structure, see *Components* at <http://bit.ly/1hZwxeJ>.

1.2 IBM Intelligent Operations Center services API

IBM Intelligent Operations Center includes a set of services that provide access to the data in the database and the capability to process the data, and act on it. The IBM Intelligent Operations Center user interface is built on top of these services. These services also provide extension points. Programmers can use the extension points to extend the IBM Intelligent Operations Center functions to develop capabilities specific to the domain and client requirements.

Figure 1-2 shows the architecture of the IBM Intelligent Operations Center solution imposed over the standard topology. For more information about IBM Intelligent Operations Center topologies, see the *IBM Intelligent Operations Center 1.5 to 1.6 Migration Guide*, SG24-8202.

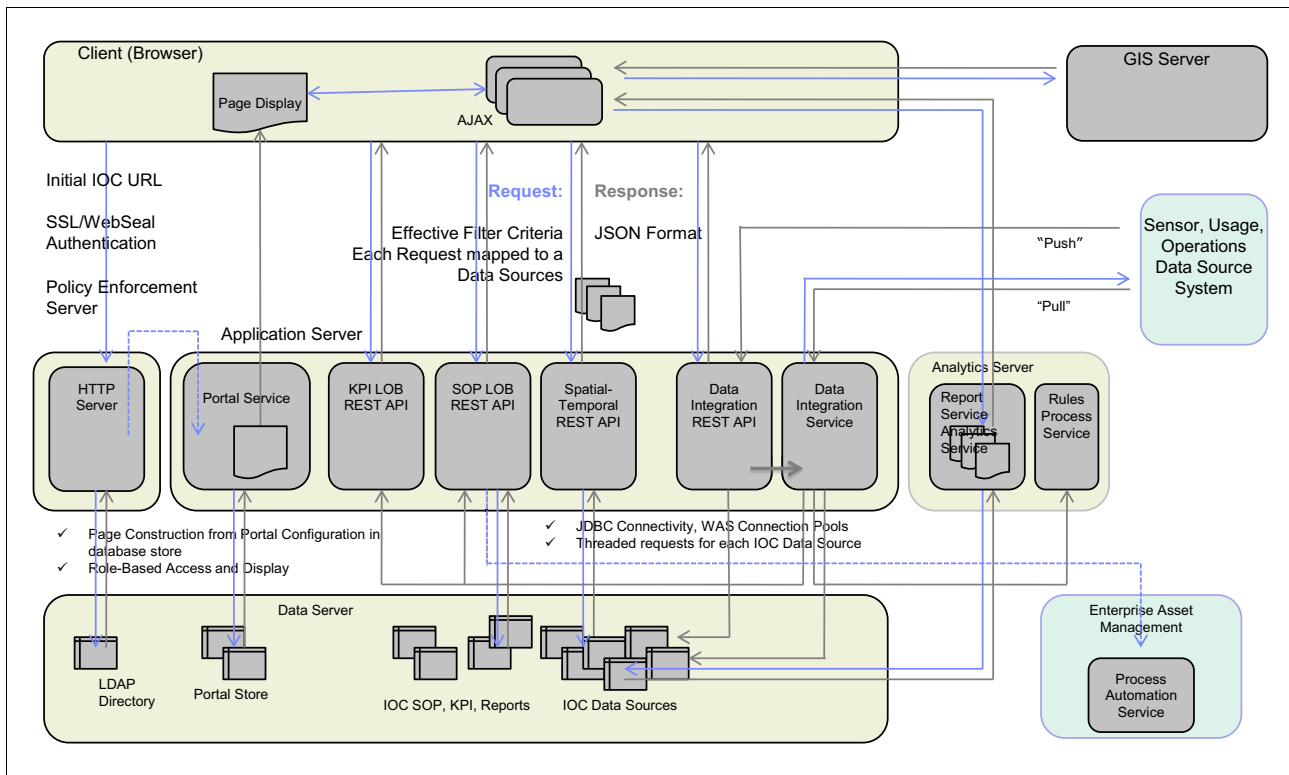


Figure 1-2 IBM Intelligent Operations Center solution request/response flow architecture

Essential integration points for IBM Intelligent Operations Center are provided by the core components: WebSphere Portal, WebSphere Application Server, IBM Business Monitor, and IBM DB2. The IBM Intelligent Operations Center data ingestion process allows you to pull information from tables and csv files. Data can also be acquired from Common Alerting Protocol (CAP) messages through a push mechanism. The Representational State Transfer (REST) APIs can be used to push information into IBM Intelligent Operations Center.

WebSphere MQ is used to process CAP messages. WebSphere Message Broker is installed but is not used in IBM Intelligent Operations Center 1.6 as part of the data ingestion process.

The services provide a standard way to access the data, both from client-side code (via the REST interfaces) and from server-side code (via the Java interface). These services have the following characteristics:

- ▶ Results are currently returned to the REST service as a JavaScript Object Notation (JSON) object (as a plain old Java object (POJO) to the Java API) but it is extensible, so other formats, such as XML, can be supported in the future.
- ▶ The REST services follow current industry coding standards for REST services. For a complete list of supported standards, see *IBM Intelligent Operations Center REST interface*:

<http://bit.ly/1hTYj2i>

For IBM Intelligent Operations Center REST interface reference documentation, see *IBM Intelligent Operations Center V1.6 REST APIs*:

<http://www-01.ibm.com/support/docview.wss?uid=swg27039957>

- ▶ The Java APIs provide a standardized way to access the data from server-side code, without writing low-level Java Database Connectivity (JDBC) code. For information about the Java API, see *IBM Intelligent Operations Center Java API documentation*:
<http://bit.ly/1hU0Md9>
- ▶ A set of standardized exceptions is also provided.

1.2.1 IBM Intelligent Operations Center services API design

The high-level design of the IBM Intelligent Operations Center service API is shown in Figure 1-3.

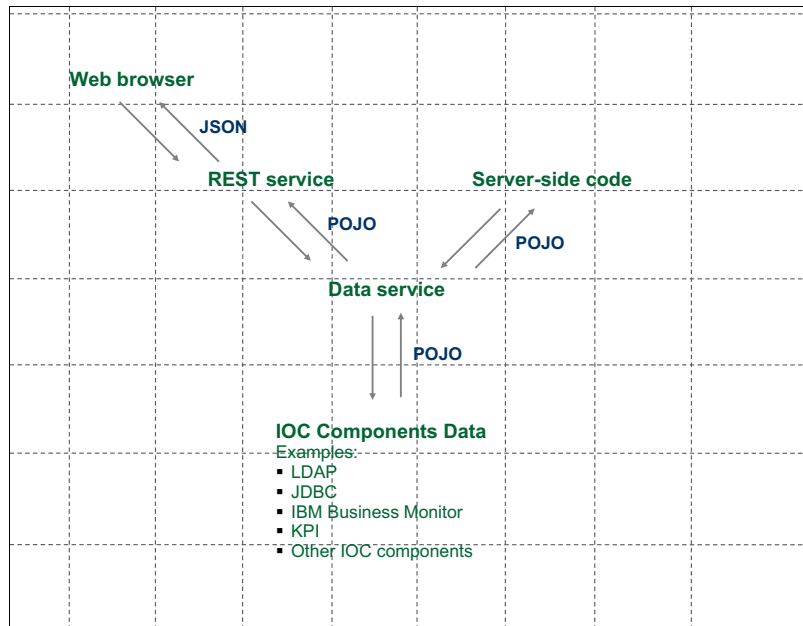


Figure 1-3 IBM Intelligent Operations Center Service API design

Design

The design allows for the function that is implemented by the data service that extends `com.ibm.ioc.rest.DataServiceImpl` to be processed either from a web browser standard REST service call, or from a server-side client. To do this, the design specifies a data service class. Because the function is now separate from the client REST service, server-side code can also access the function through the Java API, without requiring an HTTP REST call. The REST service becomes a wrapper around the data service.

In this design, the plain old Java object (POJO) classes transport the incoming parameters and any outgoing result. The same POJO classes are used throughout the design and consist primarily of attributes, getter and setter methods, and a few additional helper methods.

The data service class hides the underlying implementation that is used to interact with the actual data. The underlying implementation can be JDBC, Lightweight Directory Access Protocol (LDAP), or some other non-database API. None of the code, except for the data service, needs to directly access the underlying implementation. All the code must only access either the data service (server-side code follows this approach) or the REST service (for client-side code).

This design is implemented in the package `com.ibm.ioc.rest` in the `ioc.jar` that is available in IBM Intelligent Operations Center. The package and associated classes are described in 1.5.1, “Implementation details” on page 18.

Flow of data

A request can originate from either the web browser or server-side code. The request can be run against a database (JDBC) or other IBM Intelligent Operations Center data components, for example, LDAP or IBM Business Monitor. For simplicity, this section uses JDBC as an example. The same explanation is true for the other IBM Intelligent Operations Center data components.

When a request originates at the browser, the browser (client-side code) calls a REST service, which invokes the data service to call the JDBC layer, which in turn, returns a POJO to the data service layer. The data service layer can enhance the POJO returned. For example, it can translate to the default locale, and return it to the REST service, which wraps the returned object into a valid response (JSON object) and returns it to the web browser.

When the same request originates at the server-side code, the server-side code makes a call directly to the data service, and gets back the exact same POJO that was sent back to the equivalent REST service call. Therefore, regardless where the request originated, the same response is returned. The client-side code shown in Example 1-1 on page 9 and the server-side code shown in Example 1-5 on page 21 illustrate this point.

Therefore, the provision of a set of REST APIs and the corresponding Java APIs gives the programmer the flexibility to be able to extend functions from either the client-side code or the server-side code.

1.3 REST services

The REST services provide a set of uniform resource identifiers (URIs) to access data in IBM Intelligent Operations Center components, such as system properties and key performance indicators (KPIs).

The services can be called from any HTTP client application, and the response is sent in the form of a JSON object, which can then be parsed and consumed by JavaScript and other tools. All connections are via HTTPS.

The REST services are accessible by default in an IBM Intelligent Operations Center installation. No special configuration is necessary to enable the REST services, and they cannot be disabled.

1.3.1 Resource URIs

Each IBM Intelligent Operations Center REST resource URI has the following format:

```
https://{host}:{port}//ibm/ioc/api/{service-name}/{resource}?{parameters}
```


Use the following description to specify the URI:

`https://<ioc_ibs>/ibm/ioc/api/<component_name_in_lower_case>-service/<plural_form_of_resource>/<resource_id_to_act_on>`

Table 1-1 shows the URI identifiers and a description of their values.

Table 1-1 REST resource URI identifiers and explanation of values

Identifier	Description of value	Example
<code><ioc_ibs></code>	Web server name for the IBM Intelligent Operations Center topology. If the web server is used, there is no need to specify the port number. If the specific IBM Intelligent Operations Center deployment does not use the web server or uses a different deployment configuration, the <code><host>:<port></code> corresponding to the deployment option needs to be used.	
<code><component_name_in_lower_case></code>	Name of the service component.	datasource or authorization
<code><plural_form_of_resource></code>	Resource or service name in plural form.	datasources or authorizations
<code><resource_id_to_act_on></code>	Resource Identifier.	81

For example, the REST service to retrieve all data sources has the following format:

`https://<ioc_ibs>ibm/ioc/api/datasource-service/datasources/`

The REST service to retrieve all properties of the data source with ID 81 has the following format:

`http://<host>:<port>/ibm/ioc/api/datasource-service/datasources/81`

After a programmer is familiar with the format, a standard format, such as this one, makes it easy to use the APIs to build extensions.

Supported HTTP verbs

The IBM Intelligent Operations Center REST service supports HTTP operations that are the equivalent of database create, read, update, and delete (CRUD) operations. Table 1-2 on page 8 shows the HTTP operations and the equivalent CRUD operation.

Table 1-2 HTTP verbs supported by the REST services

HTTP verb	Equivalent CRUD operation	Type	Description
GET	Read	Safe	Returns the current state of a resource
POST	Create	Unsafe	Creates a new resource
PUT	Update	Idempotent	Updates the resource state
DELETE	Delete	Idempotent	Removes the resource

Note: Remember these important points for HTTP verbs:

- ▶ Read-only operations are *safe*.
- ▶ Read-write operations are *unsafe*.
- ▶ *Idempotent* operations are those operations that can be repeated without affecting the state of the resource.

1.3.2 REST services access

Only authorized IBM Intelligent Operations Center users can access the REST services. All authorized users can access the GET methods. Only administrative users can access POST, PUT, and DELETE methods unless otherwise noted in the *IBM Intelligent Operations Center v1.6 REST APIs* document.

An administrative user is one who is a member of at least one of the following groups: CityWideAdmin or wpsadmins.

1.3.3 Authentication

Authentication in the IBM Intelligent Operations Center portal environment is via the portal server login. When a user logs in to the IBM Intelligent Operations Center portal using their user ID and password, the security context created is passed on to the REST services invoked. Therefore, the authentication is handled automatically when the REST service access is from a portlet interface. Similarly, when the Java APIs are invoked from a portlet or servlet running in the portal, the security context is available and passed on to the service invocation.

It is not possible to use a web browser to access an IBM Intelligent Operations Center REST URL to do a GET request.

When calling the IBM Intelligent Operations Center REST APIs from a third-party application, the HTTP BASIC authentication must be used. The Authorization header needs to be set to Basic, followed by the Base 64 encoded credentials.

If the client-side code uses the Dojo libraries provided with IBM Intelligent Operations Center to make a REST call via the `dojo.xhr` class, IBM Intelligent Operations Center handles setting the `com.ibm.ioc.sessionid` HTTP header value for the client. REST requests that do not originate through this path have to add the `com.ibm.ioc.sessionid` header to the HTTP request and set its value to the JSESSIONID cookie value in order to pass the Cross-Site Request Forgery filter.

1.3.4 Client-side service invocation sample

This section uses the sample `sample_ioc_about_reststore_portlet` provided with this book. See Appendix B, “Samples provided with this book” on page 383.

This sample provides a simple example showing how to use the `RestStore` class to poll the about REST service with a base URI `/ibm/ioc/api/about-service/swtags` and it writes the product names and product versions returned on the UI.

The about service provides access to the IBM Intelligent Operations Center software inventory tags. The inventory tags provide detailed information about the software products and fixes that are installed.

To run this sample, follow these steps:

1. Import the existing project `sample_ioc_about_reststore_portlet` from the samples directory `\SAMPLES\8201PgmModel` into the IBM Rational Application Developer environment, and build the web archive (WAR) file as described in 6.6.1, “Custom portlet development environment” on page 239.
2. Deploy the WAR file on the portal server (IBM Intelligent Operations Center application server), and add the portlet to a page, as described in 6.6.2, “Deploying custom portlets” on page 242.

The sample JavaServer Pages (JSP) `RestStorePortletView.jsp` file is shown in Example 1-1.

Example 1-1 Client-side code that invokes the about service to print installed products and version number

```
<!--
/*
 * Licensed Materials - Property of IBM
 *
 * 5725-D69
 *
 * (C) Copyright IBM Corp. 2013 All rights reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 *
 * DISCLAIMER OF WARRANTIES. The following code is sample code
 * created by IBM Corporation. This sample code is warranted to perform its
 * intended function only if used unmodified. If you modify
 * this code then it is considered provided "AS IS", without
 * warranty of any kind. Notwithstanding the foregoing, IBM
 * shall not be liable for any damages arising out of your use
 * of the sample code, even if they have been advised of the
 * possibility of such damages.
 */
-->

<%@page session="false" contentType="text/html" pageEncoding="UTF-8" %>
<%@page import="java.util.*,javax.portlet.*,com.ibm.ioc.sample.about.reststore.portlet.*" %>
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet"%>
<portlet:defineObjects/>

<%
```



```

prodStore = new RestStore({
    target: "/ibm/ioc/api/about-service/swtags",
    refreshInterval: 15000,
    autoStartPolling: false
});

prodStore.on("add", function(items) {
    refreshUI();
});

prodStore.on("delete", function(ids) {
    refreshUI();
});

prodStore.on("update", function(items) {
    refreshUI();
});

prodStore.start();

when(prodStore.ready,
    function() {
        refreshUI();
    }
);
});

unload.addOnUnload(function() {
    if (prodStore != null) {
        prodStore.destroy();
    }
});
}
);
</SCRIPT>

<H1>About : Products Installed</H1>
<DIV id="<portlet:namespace/>_content" class="iocPortletContainer sampleRestStore">
</DIV>

```

Figure 1-4 on page 12 shows the portlet output displaying the installed product names and version numbers.

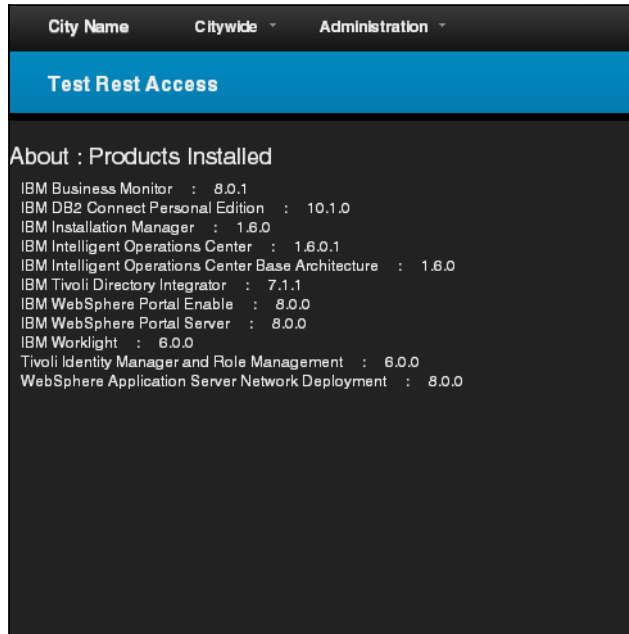


Figure 1-4 Print installed product names and versions through the about service

1.3.5 General guidelines

Consider the following guidelines when programming with the IBM Intelligent Operations Center REST services:

- ▶ Submit a GET request with a query string to search.
- ▶ A successful POST or PUT request returns the new or modified JSON object.
- ▶ Optimistic Concurrency: In a PUT request, the server compares the current last update time stamp of the resource with the value supplied in the request. The request is processed only if the time stamps match. An HTTP status code of 409 (Conflict) is returned if the time stamps do not match.
- ▶ Use path parameters to identify resources. In Example 1-2, 81 is the data source ID that is used to fetch all properties of that data source.

Example 1-2 Path parameter to identify resources

```
https://<ioc_ibs>/ibm/ioc/api/datasource-service/datasources/81
```

- ▶ For a complex search or filter criteria, use query parameters. Do not use path parameters for searching. Example 1-3 shows that the query parameter **namespace** is used to filter and retrieve all map settings for the DefaultMapSettings_IOC namespace.

Example 1-3 Using query parameters for searching

```
https://<ioc_ibs>/ibm/ioc/api/location-map-service/mapsettings?namespace=DefaultMapSettings_IOC
```

- ▶ Do not use path parameters to add new verbs to the API. For example, do *not* use the verb start following the **id** path parameter, such as `.../{id}/start`. Path parameters must only be used for identifying resources.

- ▶ Common properties: Every resource has a set of common properties:
 - `id`: Integer value. Uniquely identifies each resource.
 - `lastUpdateDate`: Long value. It is a `Timestamp` value used for a concurrency check.
- ▶ Translated properties: Text strings that are displayed in the UI, for example, a data source label, can be translated. When making a request, for example, to get the data source if it has translated values, the translated values are returned in the response. The translated value is prefixed with `i18n`. For example, for a property called `label`, the untranslated property is `label` and the translated property is `i18nLabel`. That is, `label` has the untranslated value and `i18nLabel` has the translated value.

1.3.6 Http response status codes

Table 1-3 lists the HTTP response status codes that can be received from a REST service invocation.

Table 1-3 HTTP status codes

Status code	Description
200 (OK)	Returned by a successful GET, POST, or PUT.
204 (No Content)	Returned by a successful DELETE.
400 (Bad Request)	Returned when the supplied data is not valid or data is missing (for example, input validation error).
401 (Unauthorized)	Returned when user is not authenticated to the server and the request requires authentication.
403 (Forbidden)	Returned when the user is authenticated to the server but not authorized to complete the request.
404 (Not found)	Returned by a GET or PUT request when a resource does not exist. If returning a list, return 200 and an empty list.
405 (Not allowed)	Returned when a request is made of the resource using a request method not supported by that resource. Check that the parameters specified in the call are correct.
409 (Conflict)	Returned by a PUT request when the last modified timestamp supplied by the caller does not match the current value.
500 (Internal Server error)	Returned when an unexpected error has occurred.

1.4 RestStore

The `RestStore Dojo` class is the common programmatic interface to the IBM Intelligent Operations Center REST services.

1.4.1 Purpose

The IBM Intelligent Operations Center REST services can be invoked from any UI components to query and gather information from the data sources, and other IBM Intelligent Operations Center resources.

The following factors lead to the need for an efficient and consistent way to access the REST services:

- ▶ If multiple UI code segments implement XMLHttpRequest calls to query the data, request processing and error handling might be handled differently by the various calls.
- ▶ Dynamic UI design requires that changes to data are reflected in the UI as they occur. To accomplish these dynamic updates, the REST services must be polled, which again might be handled differently by different programmers and calls.
- ▶ Efficiency considerations.

The RestStore Dojo class provides consistent access, error handling, and polling capabilities and makes the user interface more responsive and decreases the load on the server and network.

1.4.2 Benefits

The RestStore approach has the following benefits:

- ▶ Enables access to REST services data from the local data store instead of calling it directly. This approach reduces the number of back-end calls and reuses the data stored locally, which makes the user interface more responsive and decreases the load on the server and network.
- ▶ Facilitates the use of a single (common) implementation of REST services access across IBM Intelligent Operations Center.
- ▶ Provides higher efficiency because spatial queries only return the list of changed items since the last polling interval.
- ▶ Provides a simple and efficient polling mechanism implementation through events, Dojo topics, and watches, which broadcast changes when they occur to all subscribed parties.

1.4.3 Design

The RestStore design follows the standard Dojo pattern of Cache(Observable(JsonRest), Memory). The memory store is the local data store that contains the data that is retrieved from the REST calls. The JsonRest store calls the REST service. The Observable wrapper broadcasts any data changes that have occurred since the last polling interval to other widgets. The Cache wrapper reflects the JsonRest store result into the memory store.

If a change was made, for example, a delete call to the JsonRest, then that is automatically removed from the Memory store by the cache.

Note: For details about related concepts, refer to the Dojo documentation at the following links:

- ▶ Dojo/store: <http://dojotoolkit.org/reference-guide/1.9/dojo/store.html>
 - Memory: <http://dojotoolkit.org/reference-guide/1.9/dojo/store/Memory.html>
 - JsonRest: <http://dojotoolkit.org/reference-guide/1.9/dojo/store/JsonRest.html>
 - Observable: <http://dojotoolkit.org/reference-guide/1.9/dojo/store/Observable.html>
- ▶ Stateful: <http://dojotoolkit.org/reference-guide/1.9/dojo/Stateful.html>
- ▶ promise: <http://bit.ly/1jfafqM>
- ▶ topic: <http://dojotoolkit.org/reference-guide/1.9/dojo/topic.html>
- ▶ Evented: <http://dojotoolkit.org/reference-guide/1.9/dojo/Evented.html>

RestStore properties

Table 1-4 summarizes the properties of the RestStore object. These properties can be set during the initialization of the RestStore.

Table 1-4 RestStore properties

Property	Description	Example
addHeaders	Additional HTTP headers mixed into the headers property.	
autoStartPolling	Indicates whether to automatically start polling when the store is created. Default is true.	false
headers	HTTP headers to add to all HTTP requests. Default is {"Content-Type": "application/json", Accept: "application/json"}.	
id	Unique identifier for the RestStore instance. Auto generated if not specified.	PANEL_STORE
idProperty	Property to use as the identity property of the object. Default is id.	
labelProperty	Property to use as the object's label property. Default is name.	
notifyDeletedIdOnly	Indicates whether to include only the deleted object's ID when emitting delete events. Default is true.	

Property	Description	Example
queryString	Added to the REST call when polling for new, updated, or deleted objects.	
refreshInterval	Interval in milliseconds between polling cycles. Default is 0 to disable polling.	15000
sortParam	Defines the sort parameter name used in the JsonRest store.	
statQuery	Boolean property. Specifies whether STAT is making the query.	true
store	The cache store instance backed by an observable JsonRest and a Memory store.	
target	URI of the REST service that this store will be working on.	/ibm/ioc/api/filter-service/panels
timeout	REST call timeout in milliseconds. Default is to wait for the REST call to return.	60000

RestStore functions

RestStore offers some basic store functions to manipulate the data in the store. The functions are summarized in Table 1-5.

Table 1-5 RestStore functions

Function	Description
add	Creates an object in the REST service and the local store.
evict	Deletes the object from the local store; this function does not call the REST service.
get	Gets an object from the local store. Calls the REST service if the object is not in the local store.
getIdentity	Returns an object's identity (idProperty) value.
getLabel	Returns an object's label (labelProperty) value.
put	Update an object in the REST service and in the local store.
query	Query objects in the local store.
remove	Delete an object from the local store and from the REST service.
sync	Forces the store to synchronize with the REST service. Any polling is stopped before synchronizing and restarted with a new interval.
start	Starts polling. It is used when autoStartPolling is false.

RestStore promises

Promises are the programming concept that allows easier programming of non-blocking asynchronous code.¹

The promises are summarized in Table 1-6.

Table 1-6 RestStore promises

promise	Description
init	Resolved when the store is created but not ready. Use this promise with <code>autoStartPolling=false</code> to allow event handling.
ready	Resolved when the store is first loaded and ready to be used.

RestStore events

RestStore provides an event framework based on dojo/Evented to notify subscribers of changes to the data.

RestStore events support add, update, and delete events. The events are summarized in Table 1-7.

Table 1-7 RestStore events

Event	Description
add	Returns an array of objects added during the polling interval.
update	Returns an array of objects updated during the polling interval.
delete	Returns an array of object IDs deleted during the polling interval. Set the <code>notifyDeletedIdOnly</code> property to <code>false</code> to receive an array of the full item data and not just the IDs.

RestStore events support add, update, and delete events. Each event callback passes an array of items modified in the callback function. Example 1-4 shows how an add event can be added to the created RestStore.

Example 1-4 Adding event

```
sampleStore.on("add", function(items) {  
    alert("data source added!");  
});
```

RestStore Dojo topics

Similar to events, Dojo topics can be subscribed to and listened on for add, update, or delete information. The topics provide the same information and functionality as events but through a different mechanism. Developers are free to choose the most suitable mechanism for their application. Table 1-8 on page 18 shows the available Dojo topics.

¹ From Dojo documentation at <https://dojotoolkit.org/reference-guide/1.8/dojo/promise.html>

Table 1-8 Dojo topics

Topic name	Description
/ibm/ioc/reststore/add	When an object is added, RestStore publishes the added item to this topic.
/ibm/ioc/reststore/update	When an object is updated, RestStore publishes the updated item to this topic.
/ibm/ioc/reststore/delete	When an object is deleted, RestStore publishes the deleted item ID to this topic.

RestStore watches

Watches are generally used to monitor properties for changes. Based on Dojo/Stateful, watches are used to watch the number of items (count) in the store. The `count` is the number of objects in the local store, stored in the `value` property.

1.4.4 Using RestStore

The RestStore class is declared as `class com.ibm.ioc.RestStore`. It is automatically available to any Dojo code as long as the code runs in a portlet in an IBM Intelligent Operations Center theme. A new portlet can be created and added to either an IBM Intelligent Operations Center page or a new page that uses one of the IBM Intelligent Operations Center themes.

For a usage example, see 6.6.3, “Using RestStore” on page 246.

1.5 Java APIs

For reduced complexity and better performance, server-side code does not have to go through the REST service calls to access the data. Instead, the server-side code can directly call the data services to get the same POJO result, as shown in Figure 1-3 on page 5.

IBM Intelligent Operations Center provides a full set of Java APIs that correspond to the REST APIs. The Java APIs can be invoked from the server-side code.

You can access the IBM Intelligent Operations Center Java API documentation at this website:

<http://bit.ly/1hU0Md9>

1.5.1 Implementation details

Java APIs provide access to the data service layer. The data service classes extend `DataServiceImpl`, which implements the `IDataService` interface. This section describes the implementation details.

Figure 1-5 on page 19 shows the type hierarchy of `SwtagDataService`. Like all data services, it extends `com.ibm.ioc.rest.DataServiceImpl`, which implements the `com.ibm.ioc.rest.IDataService` interface, and implements methods to get a list of POJO objects that implement the `IData` interface.

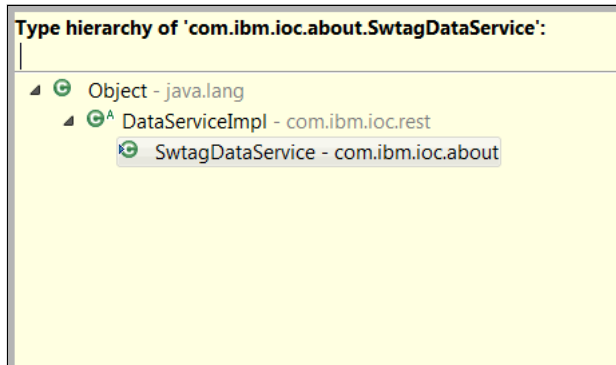


Figure 1-5 Type hierarchy of SwtagDataService

Data Service

The `com.ibm.ioc.rest.IDataService` interface is the interface implemented by the server-side REST data access services. Server-side code uses this interface to provide access to the REST data without requiring an HTTP call to the actual REST service. This interface defines the minimum required methods (create, get, update, and delete), and the exceptions that might be thrown.

The `com.ibm.ioc.rest.DataServiceImpl` class is an abstract class implementing the `IDataService` interface. All interface methods are implemented as throwing an exception. All concrete data service implementations extend this class and override any necessary methods.

POJO object

A POJO object is returned and passed around by the data service.

The `com.ibm.ioc.rest.IData` interface is the marker interface that provides a single representation for various types of POJO.

The `com.ibm.ioc.rest.DataImpl` class is the abstract class that implements the `IData` interface and defines the common REST attributes (`id`, `lastUpdateDate`, and `messages`). All concrete POJO implementations must extend this class.

1.5.2 Server-side service invocation sample

This section uses the sample projects `sample_ioc_service_examples` and `sample_ioc_service_examples_ear`. These projects are provided with this book in the directory `\SAMPLES\8201PgmModel`.

This sample is the server-side version of the sample described in 1.3.4, “Client-side service invocation sample” on page 9. It illustrates how the server-side code can invoke the data services directly.

In this example, the class `SwtagDataService` is accessed to get the list of `Swtag` elements that contain the details about product name, version, and build date. The sample code prints the product information out to the system out log and returns a JSON object containing the product name and product version.

Install and run this sample:

1. Import the samples `sample_ioc_service_examples` and `sample_ioc_service_examples_ear` into IBM Rational® Application Developer, build the application ear, and deploy it on the server, as described in “Generating and deploying the EAR file” on page 201.
2. Log in to the WebSphere Application Server 8.0 Deployment Manager console on the application server with the administrative account. You can access the console from the IBM Intelligent Operations Center portal:

Click **Administration** → **Solution Administration** → **System Administration** → **Administration Consoles** → **Application server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Lotus® Sametime® Proxy).
3. Search for `sample_ioc_service_example*`. Click the enterprise application name link. Click **Shared library references** as shown in Figure 1-6.

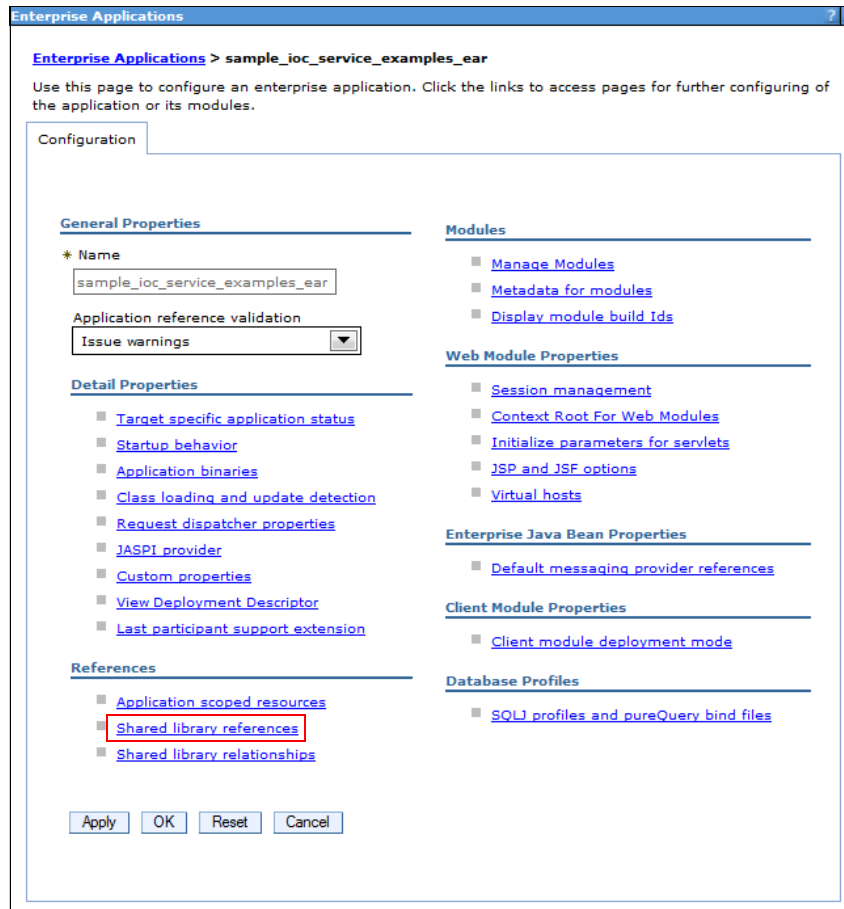


Figure 1-6 Set shared library references

4. In the next view, select the top-level enterprise application element as shown in Figure 1-7 on page 21 and click **Reference shared libraries**.

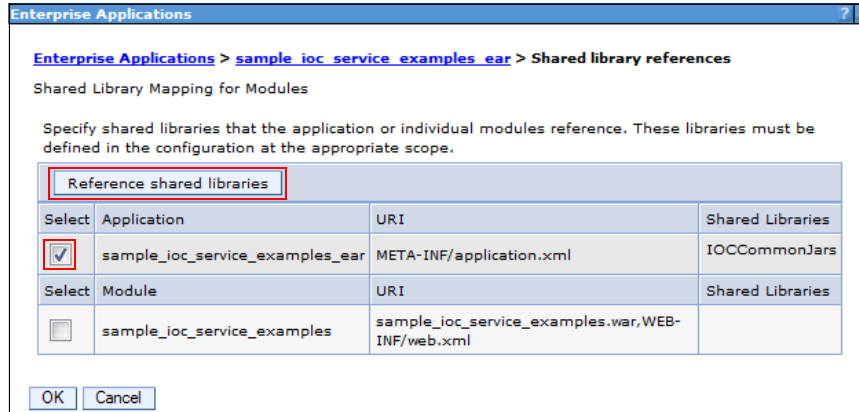


Figure 1-7 Click Reference shared libraries

5. On the next view, select **IOCCCommonJars** from the Available list box and click the right arrow to move it to the Selected list box as shown in Figure 1-8.

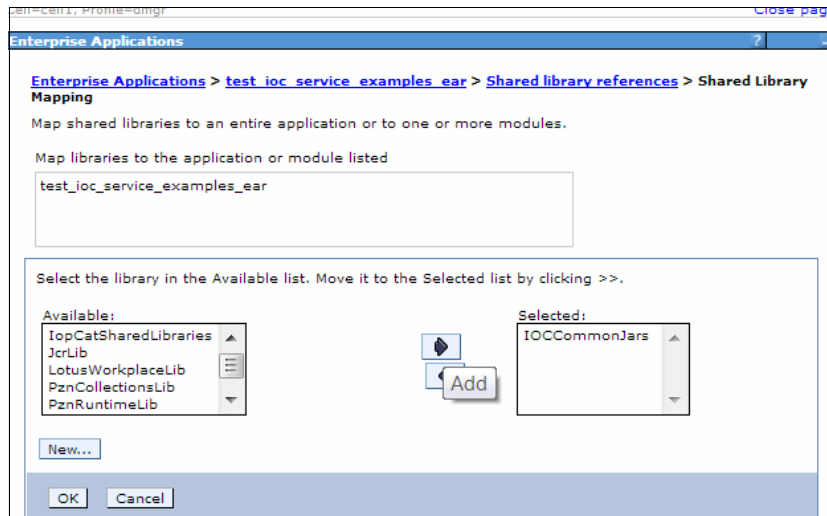


Figure 1-8 Select IOCCCommonJars

6. Click **OK** twice. A message appears reporting unsaved configurations. Click **Save**.
7. Launch the sample from the URL `https://<ioc_ihs>/sample/service/examples/swtag`.

Example 1-5 lists the sample code.

Example 1-5 Access SwtagDataService to print list of installed software products

```
package com.ibm.ioc.sample.service.examples.rs;
```

```
import java.util.Collection;
import java.util.Iterator;
```

```
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
```

```

import org.apache.http.HttpStatus;

import com.ibm.icu.util.ULocale;
import com.ibm.ioc.about.Swtg;
import com.ibm.ioc.about.SwtgDataService;
import com.ibm.ioc.rest.IData;
import com.ibm.json.java.JSONObject;
import com.ibm.ws.security.auth.Identity;

@Path("/swtag")
public class AboutExampleResource {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public Response getRecords() {
        int status = HttpStatus.SC_OK;
        JSONObject jo = new JSONObject();
        try {
            Identity identity = new Identity("wpsadmin");
            ULocale locale = ULocale.ENGLISH;
            Swtg swtg;
            SwtgDataService swtgService = SwtgDataService.singleton();
            Collection<IData> swtgLst = swtgService.get(locale, identity,
null);

            Iterator<IData> iter = swtgLst.iterator();

            while(iter.hasNext()) {
                swtg = (Swtg)iter.next();
                System.out.println(swtg.getProductName() + ":" +
swtg.getProductVersion() );
                jo.put("ProductName:" + swtg.productName,"ProductVersion" +
swtg.productVersion );
            }
        } catch (Exception e)
        {
            System.out.println(">>>>>>>> Exception " + e);
            return
Response.status(HttpStatus.SC_INTERNAL_SERVER_ERROR).build();
        }
        return Response.status(status).entity(jo).build();
    }
}

```

Example 1-6 on page 23 shows the output in the system out log.

Example 1-6 Print installed product names and versions through the SwtagDataService Java class

```
[12/30/13 12:51:06:106 EST] 00000072 SystemOut 0 IBM Business Monitor:8.0.1
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM DB2 Connect Personal Edition:10.1.0
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM Installation Manager:1.6.0
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM Intelligent Operations Center:1.6.0.1
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM Intelligent Operations Center Base Architecture:1.6.0
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM Tivoli Directory Integrator:7.1.1
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM WebSphere Portal Enable:8.0.0
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM WebSphere Portal Server:8.0.0
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 IBM Worklight:6.0.0
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 Tivoli Identity Manager and Role Management:6.0.0
[12/30/13 12:51:06:117 EST] 00000072 SystemOut 0 WebSphere Application Server Network Deployment:8.0.0
```

REST service

The server-side code can make a call to the data service layer directly and get the same return POJO object as the REST service call. This function is provided by the classes in the `com.ibm.ioc.rest` package.

The classes in the package `com.ibm.ioc.rest` listed in this section are in the JAR files in the `ioc.jar` file in the `IOCCCommonJars` shared library on the application server, and can be downloaded from the `lib` folder under `/opt/IBM/ioc/lib` on the application server:

- ▶ The `com.ibm.ioc.rest.RestResource` class is an abstract class containing common REST resource methods. All concrete REST resource classes extend this class. Extend this abstract class when creating any new IBM Intelligent Operations Center REST service.
- ▶ The `com.ibm.ioc.rest.RestUtils` class provides the common REST utilities.
- ▶ The `com.ibm.ioc.sysprop.rs.SyspropResource` class, the Sysprop REST service, extends `RestResource`.

Standard exceptions are provided as shown in Table 1-9. The exceptions extend IBM Industry Solutions standard `ISSEException` (thrown by IBM Industry Solutions products).

Table 1-9 Exceptions and equivalent HTTP response codes

Exception	Corresponding HTTP response status code
<code>BadRequestException</code>	400
<code>ConflictException</code>	409
<code>ForbiddenException</code>	403
<code>InternalServerError</code>	500
<code>NotFoundException</code>	404

Propagating messages across the layers to the UI

The `com.ibm.ioc.rest.RestMessage` class defines all the data required for a translated message. The UI can inspect and display these messages. `RestMessage` holds the `messageId`, the untranslated `messageText`, and the translated message text `i18nMessageText`.

The `java.util.List<RestMessage>` is included in `com.ibm.ioc.rest.DataImpl`. The `RestMessage` is returned with the result of the service call.

The `com.ibm.ioc.rest.ExceptionMessages` class contains `java.util.List<RestMessage>`, and it is included in any `ISSEException`. Multiple exceptions inserted will be transmitted up the chain, and they can be displayed on the UI.

Displaying messages

Library functions are provided by the `com.ibm.ioc.Library` JavaScript class.

Three messages are available in the Library function as shown in Table 1-10.

Table 1-10 Library function messages

Message	Description
<code>publishMessage(messageId, i18nMessageText)</code>	Publishes the message (ID and I18N text) to the messages topic for processing.
<code>publishRestMessages(result, defaultMessageId, defaultI18nMessageText)</code>	<code>result</code> : Zero or more success messages returned from the REST service.
<code>publishRestErrorMessages(error, defaultMessageId, defaultI18nMessageText)</code>	<code>error</code> : Zero or more error messages returned from the REST service.

Note: It is important to note that `com.ibm.ioc.Library` is a *JavaScript* class.

The UI determines the message type based on the last character of the message ID that follows the following convention: error (E), warning (W), or information (I). The current time in the preferred format of the browser is automatically added.

Messages are displayed on the upper right of the map on the Operations UI, as an IBM Dojo Extension Single Message in a scrolling div as shown in Figure 1-9. When the number of messages exceeds three, the message starts scrolling.

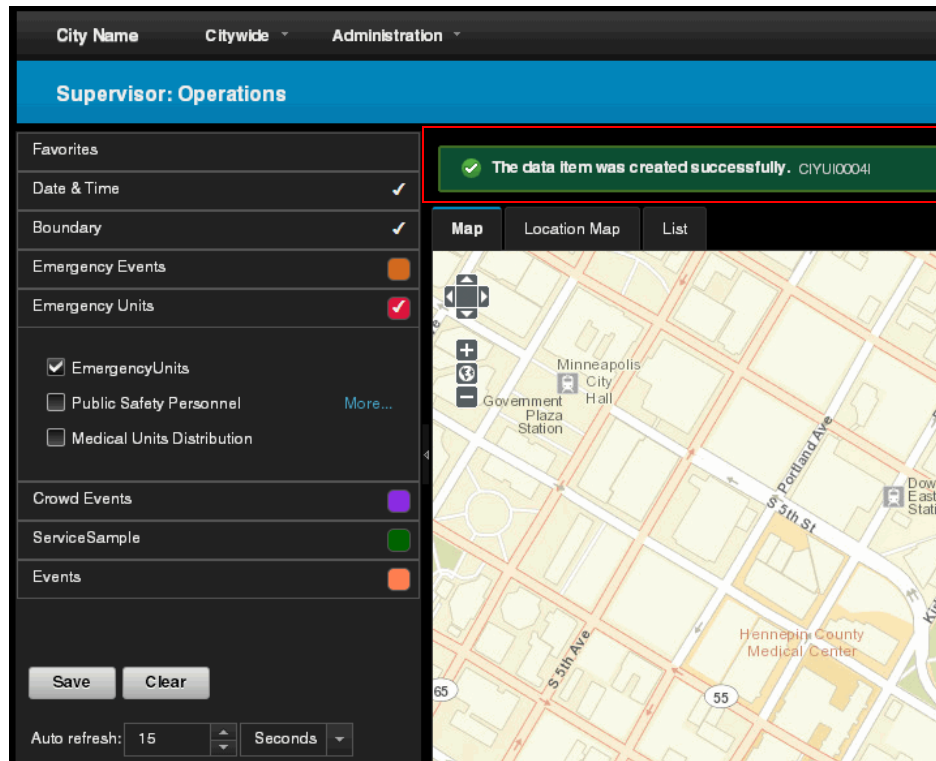


Figure 1-9 IDX SingleMessage displayed on UI in scrolling div

JDBC layer exceptions

The `com.ibm.ioc.jdbc.utils` package holds all the JDBC-related utilities.

A set of exceptions is provided as shown in Table 1-11.

Table 1-11 JDBC exceptions

Exception	Description
<code>JdbcGeneratedKeysException</code>	Unable to retrieve generated keys
<code>JdbcInsertException</code>	Unable to insert a row into the database
<code>JdbcParameterException</code>	JDBC parameter value is not valid
<code>JdbcUpdateException</code>	Unable to update a row in the database

1.6 Session persistence

Portlets persist session data, and share session information through the theme.

The session service can be used to save settings that are relevant to IBM Intelligent Operations Center users across sessions. The different IBM Intelligent Operations Center components track changes that a user makes to the user interface and various other settings and persist the changes.

The REST and data service APIs provide a mechanism to persist and retrieve session information.

The REST service base URI is `/ibm/ioc/api/session-service`.

The REST service resource URI is `/ibm/ioc/api/session-service/sessionsettings`.

The GET, POST, PUT, and DELETE HTTP methods are supported for administrative and authorized users.

Example 1-7 shows accessing the `sessionsettings` REST URI and the response to a GET request as captured with a REST client tool.

Server-side code can access the `com.ibm.ioc.session.SessionSettings` class to get and save user-specific session settings, if needed, to pass them to the corresponding UI elements.

Example 1-7 Snapshot of the response to session settings GET request

```
[7]
0: {
  uid: "admin"
  namespace: "AdminTheme"
  settings: {
    leadingPanel_style: "width: 250.4px; height: 391.4px; overflow: auto; left: 5px;
    top: 98px; position: absolute;"
    bottomPanel_open: false
    trailingPanel_open: false
    currentPageId: "10"
    currentPageUrl:
    "/wps/myportal/!ut/p/a1/pZJBb4IwFMc_C4de6auiY7s1xGiV6KYzw15MMQg1QJtSNfv2A8NRuiV7t6
```

```

a_3-vL_xVznGBei5vMhZWqFmV35tNjSG02oMGIhdE2AradE5ivCIEdaYFDC8BAUXj47_A6W7AQ1ptoT4Ct
N7PdmAbt_Uvv0wDX-1HQ-0PAavK3-Qf94H9-9Kv_hbkT6RJwA9Mec03A2aELyQ0EeI15Xqr08SM0tE7HYY
65yc6ZyYxfqMbiRC09A1k0vkwXQp7VqbyT6p6Z1xN26iwVjdvCBAMqQjuukFQfWt1rCgReFeLQCMQBAHc
T5Ss48_sw_MQYF3t20pAssukvMX0BweQLMk!/d15/d5/L3dDb0EvUU5RTGtBISEvNEpVeENBISEvWjZfOE
FMSUhBNDJJ0ENSQzBJUkcxMEDLMTEwRTA!/"
trailingPanel: true
bottomPanel: true
trailingPanel_style: "width: 15%; height: 100%; overflow: auto;"
leadingPanel_open: true
bottomPanel_style: "width: 100%; height: 15%; overflow: auto;"
currentPageParentId: "1"
leadingPanel: true
}-
id: 5
lastUpdateDate: 1384287498363
messages: [0]
}-
...
6: {
uid: "admin"
namespace: "ns_Z7_P09EHI80JOCU10IN0ES3A400S3_"
settings: {
selectedAccordionPaneId: 26
dynamicAccordionPanes: [0]
dateTimePane: {
startDate: "2013-05-07"
startTime: "T00:00:00"
endDate: "2014-04-18"
endTime: "T23:59:00"
onlyOn: {
type: "any"
}-
}-
savedFiltersPane: {
selectedSavedFilterId: -1
}-
autoRefresh: {
refreshRate: 15000
duration: 15
interval: "seconds"
}-
locationPane: {
location: "viewableArea"
}-
}-
id: 2
lastUpdateDate: 1384282652648
messages: [0]
}

```

The JavaScript class `com.ibm.ioc.SessionData` provides an easy way to use the API to store and retrieve session information as shown in Example 1-8 on page 27 and Example 1-9 on page 27.

Example 1-8 JavaScript using SessionData

```

var savedTabOrder = SessionData.getInstance().getSessionSetting(this.namespace,
'selectedTabIndex');
if(savedTabOrder && widgetConfig.tabOrder == savedTabOrder ){
    this.tabContainer.selectChild(contentPane);
}

```

Example 1-9 JavaScript updating SessionData

```

aspect.after(this.tabContainer, "selectChild", lang.hitch(this, function(){
    SessionData.getInstance().addSessionSetting({
        id: this.namespace,
        settings: { 'selectedTabIndex' :
this.tabContainer.selectedChildWidget.tabOrder}
    });
}));

```

1.7 List of IBM Intelligent Operations Center services

Table 1-12 lists the REST service resource URI, the corresponding Java data service package, and an example of where the service is used in the IBM Intelligent Operations Center product.

Table 1-12 List of IBM Intelligent Operations Center REST and data services

Service	REST base URI and resource URI	Java APIs package and classes	Description	Example of IBM Intelligent Operations Center component where it is used
About	/ibm/ioc/api/about-service	com.ibm.ioc.about	Provides access to IBM Intelligent Operations Center software inventory tags. The inventory tags provide detailed information about the software products and fixes installed.	System Administration → About
	/ibm/ioc/api/about-service/swtags	SwtagDataService		
	/ibm/ioc/api/about-service/cmptags	CmptagDataService		
	/ibm/ioc/api/about-service/fxtags	FxtagDataService		
Authorization	/ibm/ioc/api/authorization-service	com.ibm.ioc.authorization	Used to get, create, update, and delete resource authorizations.	Data source Definition → Security
	/ibm/ioc/api/authorization-service/authorizations	AuthorizationDataService		
Data injection	/ibm/ioc/api/data-injection-service	com.ibm.ioc.datainjection	Used to add, update, and cancel data items for a data source.	Map → More Actions → New Item
	/ibm/ioc/api/data-injection-service/datablocks/{data source id}/dataitems	InjectionDataService ^a		

Service	REST base URI and resource URI	Java APIs package and classes	Description	Example of IBM Intelligent Operations Center component where it is used
Data receiver	/ibm/ioc/api/data-receiver-service	com.ibm.ioc.datareceiver	Used to get, create, update, and delete data receivers. Data receivers are responsible for ingesting data records for an associated data source.	Data source Import
	ibm/ioc/api/data-receiver-service/datareceivers	DataReceiverService		
Data source	/ibm/ioc/api/datasource-service	com.ibm.ioc.datasources	The data source service is used to get, create, update, and delete data sources. Data sources are used to define the properties of the source data brought into the IBM Intelligent Operations Center.	Map (loading data sources)
	/ibm/ioc/api/datasource-service/datasources	DataSourceDataService		
Filter	/ibm/ioc/api/filter-service	com.ibm.ioc.filter	Used to get, create, update, and delete resources used for filtering data items displayed on the Map, Location Map, List, and Charts.	Configuration Tools → Filter Panel
	/ibm/ioc/api/filter-service/datasources	DataSourceDataService		
	/ibm/ioc/api/filter-service/holidays	HolidayDataService		
	/ibm/ioc/api/filter-service/panels	PanelDataService		
	/ibm/ioc/api/filter-service/ranges	RangeDataService		
	/ibm/ioc/api/filter-service/savedfilters	SavedFilterDataService		
Image	/ibm/ioc/api/image-service	com.ibm.ioc.image	Used to get a list of images installed in the IBM Intelligent Operations Center. The list of images includes the images installed by the IBM Intelligent Operations Center installer and those uploaded by a solution administrator.	ContentViewer: Location Map
	/ibm/ioc/api/image-service/images	ImageDataService		
KPI	/ibm/ioc/api/kpi-service	com.ibm.ioc.kpi.dataservices	Provides access to the key performance indicator (KPI) owning organizations, hierarchy, and values.	KPI
	/ibm/ioc/api/kpi-service/owningorgs	OwningOrgDataService		
	/ibm/ioc/api/kpi-service/kpi-tree	KpiTreeDataService		
	/ibm/ioc/api/kpi-service/kpis	KpiDataService		
	/ibm/ioc/api/kpi-service/kpi-definition	No corresponding data service class (kpi-definition)		
Location Map	/ibm/ioc/api/location-map-service	com.ibm.ioc.locationmap	Used to get, create, update, and delete categories, collections, location maps, and areas.	ContentViewer: Map
	/ibm/ioc/api/location-map-service/categories	CategoryDataService		
	/ibm/ioc/api/location-map-service/collections	CollectionDataService		
	/ibm/ioc/api/location-map-service/maps	MapDataService		
	/ibm/ioc/api/location-map-service/areaLocations	AreaLocationDataService		

Service	REST base URI and resource URI	Java APIs package and classes	Description	Example of IBM Intelligent Operations Center component where it is used
Map settings	/ibm/ioc/api/mapsettings-service	com.ibm.ioc.map.MapSettingsService	Used to get, create, update, and delete map settings. A <i>map setting</i> is a collection of base maps defined for a namespace. A base map definition includes a name, default map indicator, map type, tile server url, attribution, latitude and longitude, and zoom level.	ContentViewer: Map
	/ibm/ioc/api/mapsettings-service/mapsettings	MapSettingsService		
Notification	/ibm/ioc/api/notification-service	com.ibm.ioc.notification	Used to get, create, update, and delete notifications. Notifications provide a means for displaying alerts about events relevant to IBM Intelligent Operation Center users.	ContentViewer: Notifications
	/ibm/ioc/api/notification-service/notifications	NotificationDataService		
Security	/ibm/ioc/api/security-service	com.ibm.ioc.security	The security service returns user and group information. This service uses the WebSphere Portal Server PUMA API to obtain the user and group information. The user information is a subset of the information contained in the user profile.	ContentViewer: Security tab
	/ibm/ioc/api/security-service/groups	GroupDataService		
	/ibm/ioc/api/security-service/users	UserDataService		
Session	/ibm/ioc/api/session-service	com.ibm.ioc.session	Used to get, create, update, and delete session settings. Session settings are used to persist user interface settings across sessions for IBM Intelligent Operation Center users.	
	/ibm/ioc/api/session-service/sessionsettings	SessionDataService		
Spatial	/ibm/ioc/api/spatial-service	com.ibm.ioc.stat.service	Spatial service features resource returns the list of records for a data source that match the filter criteria selected on the filter panel. Collections resource and records subresource return all the properties for a record.	Preview Card → i (Information) icon
	/ibm/ioc/api/spatial-service/collections/{datasourceid}/records/{recordid}	RecordsDataService		
	/ibm/ioc/api/spatial-service/features/{datasourceid}	MapFeaturesDataService		

Service	REST base URI and resource URI	Java APIs package and classes	Description	Example of IBM Intelligent Operations Center component where it is used
Standard operating procedures (SOP)	/ibm/ioc/api/sop-service	com.ibm.ioc.sop	Used to get, create, update, and delete standard operating procedure resources. There are a substantial number of SOP resource URIs with a few corresponding data services. For a full list and details about how to use the URIs, see the IBM Intelligent Operations Center V1.6 REST APIs documentation.	
	/ibm/ioc/api/sop-service/reference	No corresponding data service (reference)		
	/ibm/ioc/api/sop-service/sopDefinition	SopDefinitionDataService		
	/ibm/ioc/api/sop-service/reference/0			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_UID}/versions			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_UID}/changeLog			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_VERSION_UID}/latestApprovedVersion			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_UID}/roleAccess			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_UID}/reference/{REFERENCE_UID}			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_ID}/createDraft			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DRAFT_UID}/submitForApproval			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DRAFT_UID}/discardDraft			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DRAFT_UID}/approve			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DRAFT_UID}/disapprove			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_UID}/duplicate			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_UID}/duplicate			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_DEFINITION_UID}/activityDefinition			
	/ibm/ioc/api/sop-service/sop			
	/ibm/ioc/api/sop-service/sop/{SOP_INSTANCE_UID}/changeLog			

Service	REST base URI and resource URI	Java APIs package and classes	Description	Example of IBM Intelligent Operations Center component where it is used
	/ibm/ioc/api/sop-service/sop/{SOP_INSTANCE_UID}/comment			
Standard operating procedures (SOP)	/ibm/ioc/api/sop-service/sopDefinition/{SOP_INSTANCE_UID}/roleAccess			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_INSTANCE_UID}/reference			
	/ibm/ioc/api/sop-service/sop/{SOP_INSTANCE_UID}/activity			
	/ibm/ioc/api/sop-service/sop?sopDefinitionId={SOP_DEFINITION_ID}			
	/ibm/ioc/api/sop-service/sop/sendNotification			
	/ibm/ioc/api/sop-service/sopDefinition/{SOP_INSTANCE_UID}/reference/{REFERENCE_UID}			
	/ibm/ioc/api/sop-service/activity No data service			
	/ibm/ioc/api/sop-service/sop/{SOP_INSTANCE_ID}/activity			
Standard operating procedures (SOP)	Resource URIs			
	/ibm/ioc/api/sop-service/activity/{ACTIVITY_ID}/start			
	/ibm/ioc/api/sop-service/activity/{ACTIVITY_ID}/stop			
	/ibm/ioc/api/sop-service/activity/{ACTIVITY_ID}/skip			
	/ibm/ioc/api/sop-service/activity/{ACTIVITY_ID}/complete			
	Resource URIs for start, stop If-Then-Else activity			
	/ibm/ioc/api/sop-service/activity/{ACTIVITY_ID}/startThen			
	/ibm/ioc/api/sop-service/activity/{ACTIVITY_ID}/startElse			
	ibm/ioc/api/sop-service/activity/{ACTIVITY_ID}/reference			
System properties	/ibm/ioc/api/sysprop-service	com.ibm.ioc.sysprop	Used to get, create, update, and delete IBM Intelligent Operations Center system configuration properties.	System Properties
	/ibm/ioc/api/sysprop-service/sysprops	SyspropDataService		
Internationalization	/ibm/ioc/api/i18n-service	com.ibm.ioc.i18n	Used to get, create, update, and delete translated string values for different locales.	Internationalization
	/ibm/ioc/api/i18n-service/resource	ResourceDataService		

a. In IBM Intelligent Operations Center 1.6.0.1, note the data service is InjectionDataService.

1.8 Extension points

IBM Intelligent Operations Center provides several extension points that allow IBM Business Partners and IBM domain teams to extend the base product. The following chapters include details of the available extension points and provide implementation examples.

Table 1-13 lists the principal IBM Intelligent Operations Center extension points.

Table 1-13 Principal IBM Intelligent Operations Center extension points

Extension point	Description	Section in this book
External service URL	This extension point allows you to show data from external systems or third-party applications in the IBM Intelligent Operations Center user interface. When the data in an external system cannot be brought into the IBM Intelligent Operations Center because it is not in the standard form, or due to other reasons, a service can be created to pull in data from that data source in the GeoJSON format. The URL for such a service is referred to as an <i>external service URL</i> .	3.4.1, "External service URL data sources" on page 63. 9.5, "Using an external service URL" on page 335.
Extension properties	This extension point allows you to store more complex data in special properties in the data items from the data sources.	3.4.2, "Extension properties in the resource definition" on page 63.
Routing expression	For the data that is received from a data source, routing expressions define how the data is processed by the solution. Data items that match the required expression or combination of expressions are routed to the selected destination.	3.1.1, "Data import process overview" on page 46.
Integration topic	This integration point enables the routing of data items to an external system through a routing mechanism that directs the data item fulfilling the filter criteria specified in a data source routing expression to a predefined integration topic.	9.4, "Routing data to the integration topic" on page 331.
Custom data source actions	This extension point provides the capability to add new actions to the More Actions menu in the preview card for all associated data items. This type of action can provide the function to integrate with a third-party system or to extend existing product functions.	6.3, "Creating custom data source actions" on page 219.
Content panel customization	By default, the content panel has three tabs: Map, Location map, and List. Through customization, you can change the order of this tab and add or remove custom tabs. Customizing the content panel is useful for integrating with external systems with a web user interface or to use a custom widget that interacts with or represents data from the REST services in alternative ways. The content panel is implemented through the content viewer portlet, which allows you to create new tabs and load custom Dojo widgets inside them.	6.2, "Customizing the content panel" on page 197.
Custom filter-driven results charts	This customization option allows you to add new Dojo widgets and provide alternative data visualization.	6.4, "Adding filter-driven result charts" on page 227.

Extension point	Description	Section in this book
Preview card customization	By default, all the properties for data items in preview cards are in text format. It is possible to add other types of information, such as images, links, or even video.	6.5, "Customizing preview cards" on page 233.
Custom portal pages, labels, and themes	Creating new custom <i>portal pages</i> allows you to customize the user interface for a new user role or as a way to provide new functions with custom portlets. This is a standard feature that is provided by IBM WebSphere Portal. Three custom <i>portal themes</i> are included with IBM Intelligent Operations Center: IOC Portal 8 Admin, IOC Portal 8 Theme, and IOC Portal 8 Reports. A <i>label</i> is used to contain multiple labels and pages and typically does not have any actual content. IBM Intelligent Operations Center navigation is built on two base labels: <code>ioc.Home</code> and <code>ioc.SolutionAdministration</code> . Pages created outside of these two labels will not be added to navigation.	6.9, "Creating custom solution pages" on page 257.
Custom portlets	The user interface of IBM Intelligent Operations Center can be extended by custom portlets. Developers can use the following services to optimize user experience and performance: <ul style="list-style-type: none"> ▶ RestStore JavaScript class for accessing REST services ▶ Translation service 	6.6, "Creating custom portlets" on page 239.
Translation service	The <i>translation service</i> is used to store user interface labels and other displayed text in different languages. It renders the text in the appropriate language based on the preferred user locale.	6.6, "Creating custom portlets" on page 239.
Custom help links	This customization option allows you to provide documentation for a new custom feature and makes it easily accessible from the standard Help menu.	6.7, "Adding custom help links" on page 252.
More Actions menu customization	This customization option allows you to add or remove an option in the More Actions menu on the upper-right corner of the Operations view. For example, it is possible to call a custom web form or add data into a custom portlet or widget.	6.8, "Customizing the More Actions menu" on page 255.
SOP REST service activity	The REST activity creates a REST service call to access an external entity. It can also be used to connect an SOP to another SOP running on a different IBM Intelligent Operations Center.	5.2.7, "Configuring REST service activities" on page 141.
Map and location map Dojo topics	IBM Intelligent Operations Center provides map extension points that allow developers to integrate with the map views and control the extension points. The extension points are available as Dojo topics.	7.2.4, "Map extension points" on page 283.

Extension point	Description	Section in this book
RestStore	The RestStore Dojo class provides a consistent way to access REST services, consistent error handling, and polling capabilities, makes the user interface more responsive, and decreases the load on the server and network.	1.4, "RestStore" on page 13. 6.6.3, "Using RestStore" on page 246.
Session persistence	Use the session service to save settings that are relevant to IBM Intelligent Operations Center users across sessions. The different IBM Intelligent Operations Center components track changes that a user makes to the user interface and various other settings, and persists the changes.	1.6, "Session persistence" on page 25.
Filter pane customization	The filter panel is used to filter the data source data items that are displayed on the map, location map, list, and charts. Data sources can be grouped under a category and displayed together on a filter pane. You can configure filter panes with the filter panel tool. The filter service (base URI <code>/ioc/api/filter-service</code>) can be used to get, create, update, and delete resources that are used to configure filter panes programmatically. The equivalent server-side components are also available in the package <code>com.ibm.ioc.filter</code> in the <code>ioc.jar</code> . A set of Dojo topics is also available to indicate changes that occur in the filter panel.	Video tutorials: <ul style="list-style-type: none"> ▶ Creating a DB2 data source: http://youtu.be/khnCVLq_wCE ▶ Creating a CSV data source: http://youtu.be/4fnr7w-hDE ▶ Creating a CAP data source: http://youtu.be/00mQ8phuSq
Cognos reports	Cognos reports provide additional ways to visualize the data in the IBM Intelligent Operations Center system, enabling business users to analyze the data and make business decisions.	Chapter 10, "Creating and viewing reports" on page 349.

1.9 Programming considerations for high availability

When designing and implementing custom applications to be deployed in an IBM Intelligent Operations Center highly available topology, consider the following points:

- ▶ Applications must be designed and coded to take advantage of high availability (HA). For example, applications must use state and data in the user session so that it can be replicated to the other cluster members.
- ▶ Not all application HA requirements are supported by the platform HA. For example, messaging clients (Java Message Service (JMS) and MQ Telemetry Transport (MQTT)) must implement connection retry.
- ▶ Ensure that the application is designed with clustering in mind. Careful considerations are needed to identify volatile resources and how they might behave in a clustered environment, for example, file system files and semaphores.
- ▶ The HA of resources that are external to IBM Intelligent Operations Center, such as map providers and databases, is out of the IBM Intelligent Operations Center solution provider control.

- ▶ Ensure that all file-based artifacts are available on a highly available shared file system, such as SAN, that is accessible from all nodes. Examples of file-based artifacts are data source CSV files and image files.
- ▶ Ensure that external databases used by data sources are highly available and accessible from all nodes.
- ▶ Create messaging artifacts, such as topics with the *Reliable Persistent* option. Ensure that the corresponding activation specs created are durable and shareable across the clusters.
- ▶ In-memory programming artifacts, such as cache instances, need to be in a shared global cache, accessible across clusters, with replication set as needed. All Java objects need to implement synchronization.
- ▶ When developing a new widget or portlet, use the `com/ibm/ioc/SessionData` JavaScript API to store and retrieve your own session preferences.
- ▶ If the data is acquired from a CSV file and the connection to the file is lost during the ingestion process, follow these steps to recover from the failure:
 - a. Set the polling interval of the CSV data source to zero.
 - b. If one of the columns in the CSV file is marked as an ID field during the data source creation, duplicates are ignored. So, reload the data source after placing a file with the original name in the directory on the application server.
 - c. If no ID field was designated during data source creation, check how many records were ingested into the database. Remove records already imported from the CSV file (to avoid duplicates), and replace the updated CSV file in the directory on the application server (the default location is `/opt/IBM/ioc/csv`).
 - d. Set the polling interval of the CSV data source to the desired value.
- ▶ If the data is acquired from a database (DB2 or MS SQL) and the connection to the database is lost during the ingestion process, no special handling is required. In a database node failure, the scheduler automatically redirects to the backup database node.
- ▶ For service URLs and any external REST services, the availability level of the external services determines the availability level of visualization of the data items from that service.

For information about deploying applications in an IBM Intelligent Operations Center HA environment, see the *IBM Intelligent Operations Center 1.5 to 1.6 Migration Guide*, SG24-8202.



Sample scenario

This chapter introduces the scenario on which the examples in this book are based.

Note: The solution customization and programming examples included in this book apply to IBM Intelligent Operations Center 1.6.0.1.

This chapter includes the following topics:

- ▶ 2.1, “Sample scenario overview” on page 38
- ▶ 2.2, “Planning for the event ” on page 39
- ▶ 2.3, “During the event” on page 40
- ▶ 2.4, “After the event” on page 42

2.1 Sample scenario overview

The scenario for this book is centered on a fictional city, City A.

Event organizers are planning to run a large arts festival across City A to celebrate and showcase the work of City A's talented artists. It is expected that the festival will be attended by both a large number of local residents as well as attracting an influx of visitors to the city.

The city authorities expect that this festival will place additional demands on the city's public services. They have placed certain conditions on the event organizers to ensure the safety and security of all those attending the event. The event organizers plan to use services from the private sector during the one-day festival for services, such as security marshals and first aid coverage. As part of the agreement with the city, the festival organizers must make certain information available to various interested departments in the city, such as public safety services, event planning, and health and safety.

Such large events are commonplace in cities around the world, from celebratory festivals and parades that can last many days or weeks to protests and marches that last hours.

Events, such as this one, require careful monitoring and policing to ensure the safety of all people involved. In practice, such events require collaboration across many different agencies and bodies to provide a common operational picture of what is happening across the city. Today, much of this information is both collected and disseminated among interested parties manually, leading to labor-intensive practices and suboptimal information sharing. These problems can, in extreme circumstances, have severe effects, such as overcrowding and crushing at large events. The city authorities and festival organizer want to prevent and avoid negative incidents that might cause harm to festival attendees and negatively affect City A's reputation.

In order to provide situational awareness to both the city authorities and the agencies responsible for various functions during the festival, the event organizers are planning to use IBM Intelligent Operations Center as a means to disseminate and share critical event information among a variety of public and private agencies. This system will also be used for the coordination and invocation of standard operating procedures (SOPs) to provide cross-agency responses to occurring situations.

By using IBM Intelligent Operations Center, not only will the event organizers be able to receive and collate information from a variety of sources and present it as a common operational picture to the city council; but the organizers will be able to see the availability of resources to help respond to events and incidents as they occur during the day.

The process of running such an event, however, starts long before the event date at the planning stage. City A's planning department is required to issue a permit to the festival organizers after making sure there are no conflicts that might arise with other planned public works or events running over that period of time in the same location.

In addition, the city will be keeping track of all events and incidents during the festival so they can analyze and report on the agencies' performance after the event. Recording and analyzing this information allows city authorities to understand the issues that occurred during the event and develop action plans for improvements.

2.2 Planning for the event

In planning for the event, the festival organizers need to apply to the city council for a permit to hold the event in the main city center area of City A.

This planning process takes place many months, even years, before the event and is typically initiated by the submission of forms to the city's event planning department. This information for this book's scenario includes details, such as the location of the event, the expected number of attendants, and the duration of the event.

For City A, this planning submission is entered into a third-party software system that the city authorities use to hold all events' permit applications. Information for the planning system is subsequently held in a relational database.

In order to gain a complete picture of all planned events across the city, the planning department pulls information from the event planning system into IBM Intelligent Operations Center. 3.2.2, "Use case overview" on page 52 shows how to bring data from different data sources using the standard capabilities of IBM Intelligent Operations Center. 3.2.3, "Database table data source" on page 54 describes how to import event planning information into the staging DB2 database tables.

The city council imported several other planning feeds into IBM Intelligent Operations Center from different departments, such as planned works permits from the roads and traffic department that detail any planned road work or road closures. Planning information includes event schedules for the large indoor event arenas in the area that might also be holding events on this date or might be used for the festival itself.

Bringing this cross-agency information into one common operational picture quickly allows the planner to build filters and see activities that might conflict either geospatially or temporally with other activities happening in the city on the same locations and dates.

This planning information also allows the planner to view the location of key amenities for the festival, such as the location of major thoroughfares and public transit routes to ensure that people can enter and leave the city safely.

In this scenario, City A's planner sees no conflicts and approves the permit for the festival. This approval creates a new event in the system that can be used to notify interested agencies and companies that might be affected by a large event, such as this festival.

For instance, the local public transit companies receive alerts that roads will be closed on the date of the festival, which will require them to replan bus routes for the event and potentially lay on additional services to transport people into and out of the city.

During the festival in City A, several city agencies are involved in the organization and monitoring of the event. They require well-defined and coordinated processes and activities to ensure that the festival runs smoothly and city operators can react in an orderly manner if their intervention is required. Standard operating procedures (SOPs) are a structured way to plan, monitor, and execute coordinated efforts to respond to events and incidents. 5.5, "Manage event SOP" on page 156 and 5.6, "Manage crowd incident SOP" on page 177 show how to use IBM Intelligent Operations Center SOP capabilities to plan for and manage the event and respond to crowd incidents.

In practice, event planning for any city is a complex and lengthy task that requires the coordination of multiple agencies and companies. By using IBM Intelligent Operations Center, the city council can immediately see all events, plans, and works that might affect the event and take appropriate actions or change plans to mitigate any potential issues.

2.3 During the event

During the event, the city will use the IBM Intelligent Operations Center to collate information about events and incidents occurring during the festival.

The city is interested in collecting the following information during the festival:

- ▶ Crowd sizes every 30 minutes in designated areas of the festival. These areas have controlled entry and exit for citizens and the event organizers need to limit the number of people in any one area to avoid the risk of overcrowding or crush scenarios.
- ▶ The first aid incidents that are occurring within the festival area, both at first aid stations and also those people attended by mobile first aid personnel. This information will help the festival organizers view the types of first aid happening during the festival and quickly see any trends or patterns occurring in areas that might need the organizers to intervene.
- ▶ The city will also bring in information from existing city systems, such as the closed-circuit television (CCTV) network and the number of citizens riding on the city's public transit network.

3.2.4, “CSV data source” on page 55 explains how a comma-separated value (CSV) data source is used to receive crowd size and first aid event data. This information comes from both a CSV file and a mobile application through the Representational State Transfer (REST) API described in Chapter 8, “Developing mobile applications” on page 299.

8.2, “Use case overview” on page 302 describes a mobile application used by operators onsite to report the situations that require first aid as they occur. Operators can also see the situations happening in the surrounding areas using a map view in their mobile device.

4.2, “Use case overview” on page 73 describes how to use IBM Intelligent Operations Center key performance indicator (KPI) capability to enable the city mayor to monitor events with a large crowd gathering every hour.

2.3.1 Keeping track of crowd sizes

The city will keep track of crowd sizes receiving estimates of crowd size every 30 minutes from a number of *spotters* located around the festival.

In specific cases, modern video analytics can be used to estimate crowd sizes using existing CCTV feeds, but City A does not yet have this technology deployed and it is still using manual capabilities.

Each spotter transmits the crowd size estimates to the festival control room every 30 minutes (see 3.2.4, “CSV data source” on page 55). Spotters enter the information into a simple web form that allows the operator to record the reporting area and the estimated crowd size into the system. This information is subsequently consumed into the IBM Intelligent Operations Center and updates the crowd size polygon for a specific area. See the *Creating a CSV data source* video tutorial at <http://youtu.be/4fnmr7w-hDE>.

While IBM Intelligent Operations Center collects this information over the day, the operators can quickly view how the crowd size is fluctuating for each controlled area of the festival, and spot any sharp rises or decreases in an area that might in itself indicate a wider issue.

The operators can also see through the IBM Intelligent Operations Center dashboard where warning thresholds might be triggered automatically based on the crowd size becoming too large. Business rules in IBM Intelligent Operations Center evaluate the information received for different zones against prescribed values for warning and critical setup levels for each zone. These limits can easily be changed by authorized users, to allow for changes in the situation, for example, an inclement weather forecast affecting the festival. In this scenario, the event situation commander might decide to lower the crowd size thresholds.

If a crowd size warning is triggered, a standard operating procedure can either be triggered automatically by the system or manually applied by the operator, to take actions to mitigate the crowd size and ensure that it does not reach a critical level (see Chapter 5, “Driving standard operating procedures” on page 109).

Such actions might tell security marshals to temporarily close the area and issue messages over social media to inform citizens of areas of the festival that are less busy and that might have more space for them to attend.

Beside the operational views used by the city operations staff and event organizers, the crowd size information for a particular area also drives a set of key performance indicators (KPIs) that are used in a set of management summary pages. This dashboard quickly gives the event organizers a simple *Red/Amber/Green* view across the whole festival, not only for the crowd sizes but other key metrics, such as the number of incidents, number of tickets sold, and so on (see Chapter 4, “Tracking key performance indicators” on page 69).

The event organizers are also using a crowd size prediction model for the festival to start to predict crowd sizes for the areas based on both the number of persons entering the zones during the day and the meteorological conditions during the event. This prediction will help operators to spot potential areas of risk before they occur and start to take preemptive actions to stop any issues arising.

2.3.2 Keeping track of first aid incidents

Beside keeping track of crowd sizes in the festival, city authorities require the event organizers to report all first aid incidents that occur during the festival, either at the first aid stations that are set up in the crowd areas or mobile events handled by the mobile first aid teams.

First aid staff on the ground must capture this information and report it to the festival control center. Then, this information is presented back to festival operations staff, festival management, local city council, and local hospitals. Chapter 8, “Developing mobile applications” on page 299 describes the implementation of this use case.

The first aid staff will capture the injury information on mobile smartphones, allowing them to quickly select from a number of drop-down options to categorize the severity and type of event and the actions that were taken to respond to the injury.

This information will drive operational views, showing the number of first aid incidents over the festival area and allowing operators to apply both geospatial and temporal filters to this information to look for any clustering of events happening at a given point in time.

Preset KPIs are also populated by the information received, and any issues can easily be spotted by thresholds being breached for different geographical zones of the festival or for the severity of types of events. Chapter 4, “Tracking key performance indicators” on page 69 describes the implementation of this use case.

In case of any serious incidents that require additional intervention, such as an injury requiring hospital admission, the festival operator can use IBM Intelligent Operations Center to locate the nearest first aid staff or ambulance and direct the first aid team to this location. The operator can also inform the hospital before the injured person arrives, allowing them to prepare for admission to the emergency department. Chapter 7, “Visualizing information with maps” on page 269 includes a use case to find the location of the nearest emergency unit.

2.3.3 Responding to incidents

Beside collating, processing, and disseminating information from the festival organizers, IBM Intelligent Operations Center will be used by the local public safety services to look for and respond to incidents as they occur during the festival.

The local public safety team using IBM Intelligent Operations Center has access to additional feeds and information for their use during the festival and therefore can layer additional information into their operational view. They can use the live feed from the CCTV cameras located around the city to be able to view the live crowd sizes in areas and correlate this information with the reported estimates from the festival organizer’s spotters. 6.3, “Creating custom data source actions” on page 219 shows how to add a dummy CameraFeedAction.

The public safety team can also view additional information, such as the location of the event safety contractors, the local police officers, and special forces on the ground during the festival. They can also bring in information, such as the number and type of arrests made during the festival, to look for any obvious geospatial trends that might warrant further investigation during the event. See 9.5, “Using an external service URL” on page 335, for a description of how to use a service URL on the data source REST service, to pull in data from external systems, for visualization on the map or list views.

If the crowd size exceeds a predefined threshold, a standard operating procedure needs to be triggered to manage any possible emergencies. The city’s asset management system runs on IBM SmartCloud® Control Desk; therefore, the IBM Intelligent Operations Center SOP must trigger a job plan in the external system. See 9.2, “Integration with IBM SmartCloud Control Desk through SOP automation activity” on page 322 for a description of the integration with IBM SmartCloud Control Desk. See 5.6, “Manage crowd incident SOP” on page 177 for how to use a data source routing expression to trigger an SOP when the crowd size exceeds a predefined threshold.

Using the SOP capability within IBM Intelligent Operations Center allows the operator to quickly step through the predefined steps, with all actions taken by the relevant agencies, assigned tasks being tracked and recorded during the event for reporting and playing back later.

2.4 After the event

During the event, all information and events that occurred were captured in IBM Intelligent Operations Center, which makes the solution an extremely useful platform to analyze the overall performance of the city during the festival. The solution is also used to debrief the agencies involved and to understand where shortcomings were observed and what lessons might be learned for the future.

Using the extensive reporting capabilities within IBM Intelligent Operations Center, the city is able to create a number of business intelligence reports to analyze all aspects of the festival, such as the crowd sizes, the number of security staff, and the number of first aid incidents throughout the day and their severity and location in the festival. Chapter 10, “Creating and viewing reports” on page 349 describes how to use IBM Cognos Business Intelligence to create Cognos reports that use data from IBM Intelligent Operations Center data sources and are displayed in IBM Intelligent Operations Center.

Using this information, the city and the festival organizers can learn what can be done better next time, and provide a safer and more efficient festival experience for everyone involved.



Integrating with data sources

Bringing data into the IBM Intelligent Operations Center solution is a critical process. The information can come in different formats and from different types of sources. It is important to understand key considerations when planning to connect to a new data source.

This chapter includes the following topics:

- ▶ 3.1, “The data import process” on page 46
- ▶ 3.2, “Configuring data sources” on page 49
- ▶ 3.3, “Using APIs to send data into the system” on page 58
- ▶ 3.4, “Extension points” on page 62
- ▶ 3.5, “Troubleshooting data sources” on page 64

Note: For information about the sample data for the data sources described in this chapter, see Appendix B, “Samples provided with this book” on page 383.

3.1 The data import process

This chapter describes the data import process in IBM Intelligent Operations Center, its architecture, and the overall flow.

This section includes the following topics:

- ▶ The data import process overview
- ▶ Extension points

3.1.1 Data import process overview

The data import process in IBM Intelligent Operations Center is implemented in a component called the data receiver. The *data receiver* consists of the data source configuration and run time.

The data source configuration is created by the system administrator using the data source configuration tool. It includes data source connection information, properties of the data items imported from the data source, user security settings, routing expressions, and appearance settings (marker icon, fill colors, and so on). Figure 3-1 shows step 1, Acquire, in the data source configuration tool.

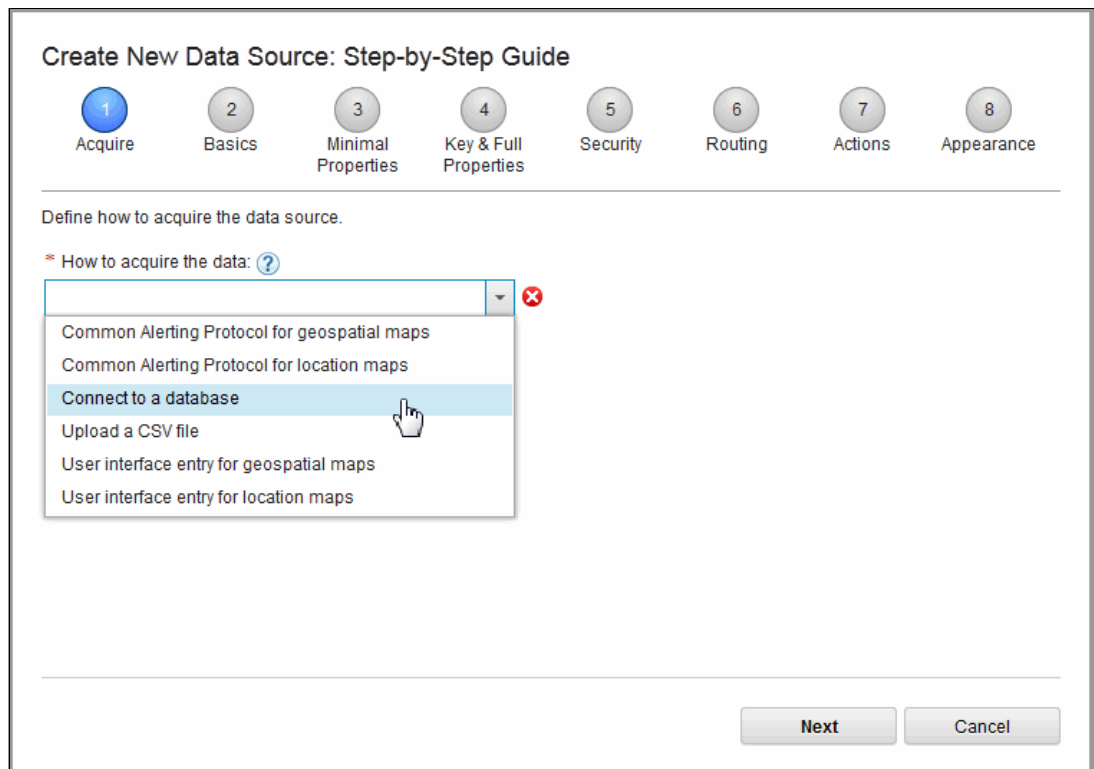


Figure 3-1 Data source configuration tool - Step 1 Acquire

The data source configuration is stored in the system database and it is used by the data receiver runtime component to import the data.

When the data source is configured to import data from external database tables or comma-separated value (CSV) files, the runtime component is triggered by the scheduler each time that it needs to start the data import process. The scheduler is configured as a polling interval in the data source configuration and specifies how often to check for new data and update the system database.

The high-level architecture of the data import mechanism in IBM Intelligent Operations Center can be represented by the diagram shown in Figure 3-2.

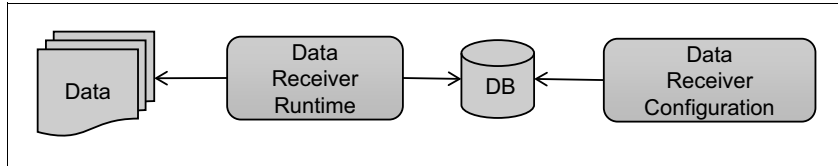


Figure 3-2 High-level architecture of IBM Intelligent Operations Center data import process

Another way to get data into IBM Intelligent Operations Center is to use Java or Representational State Transfer (REST) APIs or send a Common Alerting Protocol (CAP) message to the system. In this case, the data receiver runtime component is triggered by the API call or a message-driven bean (MDB) when receiving CAP messages, instead of by the scheduler.

The system administrator can choose from one of the predefined data source types: DB2 or SQL Database Table, CSV file, or incoming CAP messages. For more information about each data source type, see 3.2, “Configuring data sources” on page 49.

When all the configuration parameters are defined and the administrator clicks OK on the final step of the data source configuration tool, the configuration data is sent to the data receiver service to store this information in the system database.

Data records that are coming from data sources are called *data items*. During the data source configuration, the system administrator can create routing expressions that define the criteria by which data items are sent for further processing to another IBM Intelligent Operations Center component, such as key performance indicators (KPI), standard operating procedures (SOP), or an integration topic, or just save the data items into the system database.

After the data source configuration is complete, the data receiver run time is ready to start processing. This process is illustrated in Figure 3-3 on page 48.

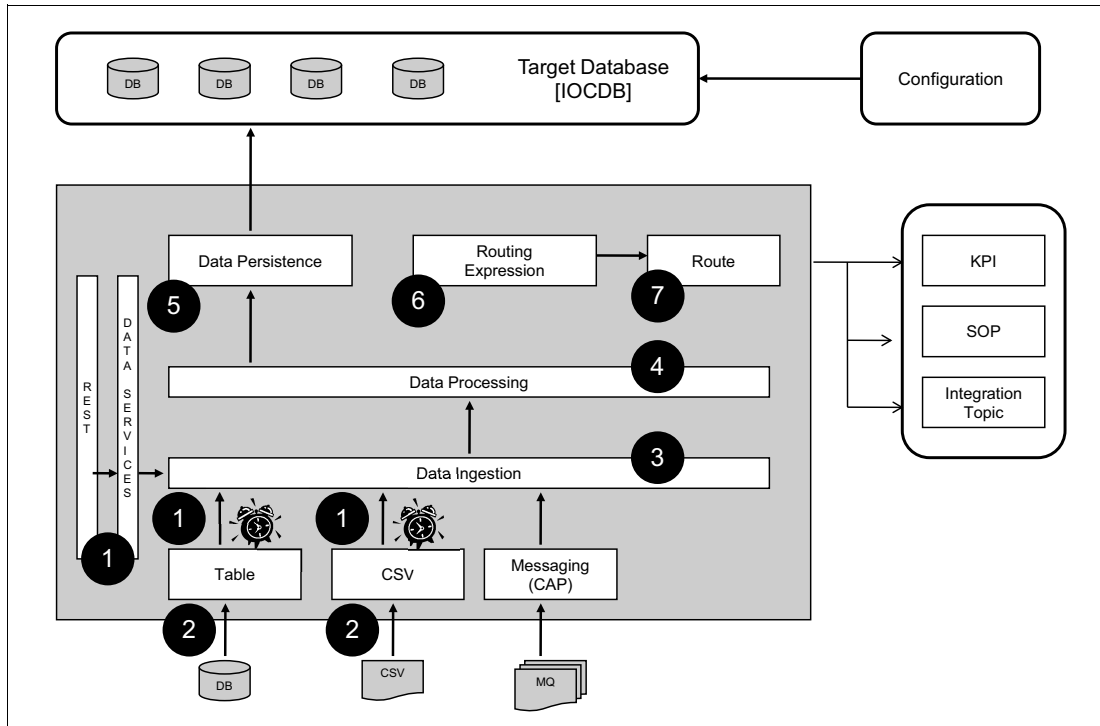


Figure 3-3 Data import flow overview diagram

Figure 3-3 shows the following numbered steps:

1. The polling interval specified in the data source configuration expires or data is sent in a CAP message or through API calls.

Note: Data items can be pushed into the system using REST or Java APIs (see more details in 3.3, “Using APIs to send data into the system” on page 58). This approach allows you to send data directly to the data source without waiting for the scheduler to pull the data items.

2. The data receiver run time starts pulling the data from the specified source and imports it into IBM Intelligent Operations Center. For example, if the source is a table in a DB2 database, the data receiver runs a SELECT query to get all the records with the last update time greater than the last consumed one.
The CAP message is received from the MDB as an XML file. If data is sent into the system using a REST or Java API, this step is skipped.
3. While the data is pulled into the system, data ingestion applies the filtering criteria defined by the routing expressions of the data source configuration. If the incoming data item meets any of the routing expressions criteria, it is pulled into the system for further processing.
4. The filtered data is then sent to data processing, which splits the data into sets of 1,000 data items.

Note: By default, data processing processes 1,000 data items in one time. If the amount of data is larger, it splits the data into chunks of 1,000 data items to avoid performance and resource issues. However, the size of each chunk can be configured using the system property `WorkLoadSize`. To access the system properties, navigate to **Administration** → **Solution administration** → **System Administration** → **System Properties**.

5. The data items are stored in the system database in the target table for the corresponding data source.
6. The data items are prepared to be routed to BM Intelligent Operations Center components, such as key performance indicators (KPIs), an integration topic, or a standard operating procedure (SOP).
7. The data items are routed according to the criteria defined by the routing expressions for that data source.

Note: There are five destination options for the routing expressions in the data source configuration: Intelligent Operations Center, Key Performance Indicators, Standard Operating Procedures, Correlation Topic, and Integration Topic. For more information about each destination option, see *Routing data from your data source* at this website:

http://pic.dhe.ibm.com/infocenter/cities/v1r6m0/topic/com.ibm.ioc.doc/use_datasource_routing.html

However, there is a difference between the destination option Intelligent Operations Center and the other destination options. Intelligent Operations Center defines which data items will be stored in the system database during filtering in Step 3, while all other options are applicable *only* to the data items that are stored in the database.

By default, if the Intelligent Operations Center destination option is not specified, all data items are stored in the target table. If the option Intelligent Operations Center is present, it allows you to filter out some data items before routing to other destinations.

For example, if the criterion defined in the routing expression specifies that data items with `Property 1 >= 100` will be routed to Intelligent Operations Center, all the other routing expressions after it apply *only* to the data items with `Property 1 >= 100`.

3.2 Configuring data sources

This section includes the following topics:

- ▶ Key considerations when planning to import data into IBM Intelligent Operations Center
- ▶ Use case overview
- ▶ Database table data source
- ▶ CSV file data source
- ▶ CAP messages data source
- ▶ User interface entry data source

3.2.1 Key considerations when planning to import data into IBM Intelligent Operations Center

IBM Intelligent Operations Center can import information from a large variety of different sources of data. This can be easily accomplished by defining the data source using the data source configuration tool. To access the data source configuration tool, navigate to **Administration** → **Solution administration** → **Configuration Tools** → **Data Sources**.

A data source configuration specifies the following information:

- ▶ The type of data source
- ▶ The location of the data
- ▶ The connection information required to access the data at the specified location
- ▶ How often to look for new updates
- ▶ How to map the source data to the system's target tables (for example, what source columns contain the data source properties)
- ▶ What users and groups can access the data
- ▶ Where to route data items as they are imported
- ▶ What visual styles need to be associated with the data

There are four sources of data that can be used to get data into IBM Intelligent Operations Center:

- ▶ A table in a DB2 database
- ▶ Text file in CSV format
- ▶ Common Alerting Protocol (CAP) messages
- ▶ The user interface entry data by selecting **More Actions** → **New Item** from the operator's view

Another way to send data into the system is using the REST or Java API. These APIs are available for any data source independently of its type.

Having all these options to bring data into IBM Intelligent Operations Center provides great flexibility, but it is important to understand the differences between the options. To compare the available data ingestion methods, see Table 3-1 on page 51.

Table 3-1 Data ingestion methods comparison

Data ingestion method	Data ingestion type	Time required to get data item into the system	Is programming required to get the data into the system?
DB2 database table	Pull	Minimum 1 minute.	No
CSV file	Pull	Minimum 1 minute.	No
CAP message	Push	The CAP message can be received immediately, but it can take up to 1 minute to bring it into the system.	Yes. Message needs to be sent to the CAP.IN.Q queue in IBM WebSphere MQ. See 3.2.5, "CAP data source" on page 56. However, the configuration does not require programming.
REST or Java API	Push	Immediately.	Yes. Must implement REST or Java API call for any configured data source. See 3.3, "Using APIs to send data into the system" on page 58.

The user interface entry data is published through a REST API call. Therefore, the characteristics for this data source are the same as the ones for REST or Java APIs, except that there is no need for programming because it is available with the product. For more details, see 3.3, "Using APIs to send data into the system" on page 58.

When solution administrators create a data source configuration, they define minimal, key, and full properties:

- ▶ Minimal properties

They are the set of properties that are needed to place items on a map, to run filter-driven charts, and for other functions. The minimal properties are required in the data source configuration. They are defined in the step 3, Minimal Properties tab of the data source configuration tool. They include location, time range, and last modified time. This set of properties provides information that is required to filter and position data items on the map.

Note: The following list shows all of the minimal properties:

- ▶ Name
- ▶ Location
- ▶ Timezone offset
- ▶ Start date time
- ▶ End date time
- ▶ Last update date time

Date and time properties and timezone offset can be omitted. If they are omitted, the following defaults are used:

- ▶ Start date time defaults to current date time
- ▶ End date time defaults to current date time
- ▶ Timezone defaults to Greenwich mean time (GMT) timezone

Location values must be specified in well-known text (WKT) format, which is a text markup language for representing vector geometry. Another option is to choose two properties from the source of the data and map them to the Latitude and Longitude values.

For more information about WKT, see *Well-known text (WKT) representation* at this website:

<http://bit.ly/1kXBqdR>

- ▶ **Key properties**

Key properties are optional and are used for additional filtering and display. These properties are displayed on the List tab of the content viewer portlet and on the data item preview card on the map.

Any property from the full properties list can be assigned key property status on the Key & Full Properties tab of the data source configuration tool by changing the value of the Key property field to **Yes**.

- ▶ **Full properties**

Full properties are optional and provide additional information about the data item. By default, all data source properties are full properties, which means all the properties that are not minimal are automatically full properties. The full properties are shown on the preview card when the user clicks *More information*.

For more information about preparing data before importing it, see *Guidelines on preparing data for the solution* at this website:

<http://bit.ly/QvhWCf>

For step-by-step instructions about configuring a data source, see *Configuring a data source* at this website:

<http://bit.ly/1h1kpoz>

3.2.2 Use case overview

In City A, different city agencies and festival organizers use IBM Intelligent Operations Center to plan the event, monitor situations during the event, and analyze the event after it ends. All these activities require different kinds of data to be imported into the system.

This use case shows how to bring data from different data sources using the standard capabilities of IBM Intelligent Operations Center.

Actors

In this use case, the IBM Intelligent Operations Center administrator defines the requirements for external data sources and configures the system to import data from the external data sources.

This use case uses external software systems, which are different kinds of systems that are managed by multiple organizations that need to provide data to a centralized situation center.

Desired outcome

The desired outcome of this use case is data that is imported into IBM Intelligent Operations Center and ready to be used by the system operators and City A council analysts.

Setting up the development environment

The following development tools are used in this use case:

- ▶ IBM Intelligent Operations Center data source configuration tool. This tool can be accessed from the IBM Intelligent Operations Center portal. Navigate to **Administration** → **Solution administration** → **Configuration Tools** → **Data Sources**.
- ▶ IBM WebSphere MQ Explorer. This tool is installed by default on IBM Intelligent Operations Center analytics server and is used to send test CAP messages in 3.2.5, “CAP data source” on page 56.
- ▶ DB2 client application. There are multiple tools that can connect to a DB2 database and run queries against it. Developers can use any of the following tools:
 - IBM Data Studio: This tool provides an integrated environment for database development and administration. IBM Data Studio can be downloaded at no charge from this website:
<http://www-03.ibm.com/software/products/en/data-studio/>
 - DB2 command line processor: This tool can also be used to run queries against the IBM Intelligent Operations Center database.
 - Data Source Explorer view in IBM Rational Application Developer.

The DB2 command line processor is used in the example scenario in this section.

Implementation summary

The summary implementation steps for this scenario are:

- ▶ Create a DB2 data source and import the event planning information (planned events and permit applications). This data comes from third-party software and it is first stored in a staging DB2 database.
- ▶ Create a CSV data source with emergency situation events, such as crowd size reports and first aid incidents. The data comes from CSV files and is used by the custom mobile application in Chapter 8, “Developing mobile applications” on page 299.
- ▶ Create a CAP data source that receives events with information about the location of Public Safety Personnel. This data comes into the system as CAP messages.

3.2.3 Database table data source

According to the scenario, the staging DB2 database tables will hold information about event planning. This is one of the approaches that can be taken during implementation if it is required to integrate with another system. The DB2 database in this case contains staging tables, which will store the data that is cleared and ready to be imported into IBM Intelligent Operations Center. Data clearance is important when planning for integration with third-party applications because different systems can have different formats for dates, time, location, and might have important properties, such as a unique identifier. All these properties need to be converted into the formats that are supported by IBM Intelligent Operations Center and its minimum required properties (minimal properties) described in 3.2.1, “Key considerations when planning to import data into IBM Intelligent Operations Center” on page 50.

Note: If the plan is for the staging database tables to receive updates, and, in turn, update data items in the system database, the LastUpdateDateTime column must be present and populated. If there is no LastUpdateDateTime column in the source table, there is no way to check for updates for each data item.

Connecting to DB2 database table

To create a staging table with sample data for the data source, follow these steps:

1. Log in to the IBM Intelligent Operations Center data server as db2inst2.
2. Create a sample database called SAMPLE to hold the sample staging table. Run the following command from the command-line interface (CLI) on the data server:

```
db2 CREATE DATABASE SAMPLE AUTOMATIC STORAGE YES
```
3. Create a new database schema named SAMPLE:

```
db2 CREATE SCHEMA SAMPLE
```
4. Create a table named SAMPLE.PERMITSANDEVENTS for the sample data:

```
db2 CREATE TABLE SAMPLE.PERMITSANDEVENTS (ID CHAR(20), LOCATION VARCHAR(800),  
CATEGORY CHAR(50), STATUS CHAR(50), ADDRESS CHAR(254), REQUESTOR_INFO CHAR(50),  
DESCRIPTION VARCHAR(500), START_DATE_TIME TIMESTAMP, END_DATE_TIME TIMESTAMP,  
LAST_UPDATE TIMESTAMP, TIMEZONE_OFFSET INTEGER)
```
5. Add sample data into the newly created table:

```
db2 INSERT INTO "SAMPLE"."PERMITSANDEVENTS" ("ID", "LOCATION", "CATEGORY",  
"STATUS", "ADDRESS", "REQUESTOR_INFO", "DESCRIPTION", "START_DATE_TIME",  
"END_DATE_TIME", "LAST_UPDATE", "TIMEZONE_OFFSET") values('3827',  
'POINT(-93.256141 44.973102)', 'Planned event', 'Pending', '11th Avenue S,  
Stadium Parking Lot', 'Stadium Management', 'Parking lot maintenance work',  
'2013-12-03 08:00:00', '2013-12-03 16:00:00', '2013-10-03 08:00:00', '0')  
  
db2 INSERT INTO "SAMPLE"."PERMITSANDEVENTS" ("ID", "LOCATION", "CATEGORY",  
"STATUS", "ADDRESS", "REQUESTOR_INFO", "DESCRIPTION", "START_DATE_TIME",  
"END_DATE_TIME", "LAST_UPDATE", "TIMEZONE_OFFSET") values('2837',  
'POINT(-93.258501 44.97248)', 'Planned event', 'Pending', '818-844 S 5th  
Street', 'City council', 'Road reconstruction', '2013-11-27 08:00:00',  
'2013-12-01 16:00:00', '2013-10-05 08:00:00', '0')  
  
db2 INSERT INTO "SAMPLE"."PERMITSANDEVENTS" ("ID", "LOCATION", "CATEGORY",  
"STATUS", "ADDRESS", "REQUESTOR_INFO", "DESCRIPTION", "START_DATE_TIME",  
"END_DATE_TIME", "LAST_UPDATE", "TIMEZONE_OFFSET") values('3746',  
'POINT(-93.2576 44.973952)', 'Permit applications', 'Approved', '900 S 5th
```



```
Street', 'UFO foundation', 'International UFO Congress', '2013-11-15 08:00:00',
'2013-11-20 16:00:00', '2013-09-09 08:00:00', '0')
```

```
db2 INSERT INTO "SAMPLE"."PERMITSANDEVENTS" ("ID", "LOCATION", "CATEGORY",
"STATUS", "ADDRESS", "REQUESTOR_INFO", "DESCRIPTION", "START_DATE_TIME",
"END_DATE_TIME", "LAST_UPDATE", "TIMEZONE_OFFSET") values('3253',
'POINT(-93.256506 44.972798)', 'Permit application', 'Approved', '11th Avenue
S, Stadium Parking Lot', 'City council', 'Kids/Parents night out', '2013-10-03
08:00:00', '2013-10-03 23:00:00', '2013-09-03 08:00:00', '0')
```

Now, the staging table is created and populated with the sample data.

After creating a staging table and populating it, see the *Creating a DB2 data source* video tutorial for instructions to create a new data source:

http://youtu.be/khnCVLq_wCE

3.2.4 CSV data source

The CSV data source is used to receive crowd size and first aid events data. This information comes from both a CSV file on the file system in the application server and the mobile application through the REST API described in Chapter 8, “Developing mobile applications” on page 299.

The CSV file must meet the following requirements:

- ▶ The first line of the CSV file must contain a list of the property names separated by commas (no diameters).
- ▶ Each subsequent line must contain an identical number of values to match the property names in the first line.
- ▶ No extra commas at the end of any line are allowed.
- ▶ For any date field (defaults of StartDateTime, EndDateTime, and LastUpdateDateTime if present), there must be no quotation marks, slashes, or other punctuation marks.
- ▶ If a column contains location information in WKT format, it must be surrounded by quotation marks as shown in Example 3-1.

Example 3-1 Example of a line in WKT format that can be used in a CSV file

```
"LINestring(-92.744843 44.366516, -92.706799 44.395773, -92.659781 44.375156,
-92.730798 44.370514, -92.742255 44.465424)"
```

- ▶ The default date format for the CSV file is YYYY-MM-DD HH:MM:SS (for example: 2011-05-27 13:25:14). This format can be changed during the data source creation when defining minimal properties on the Timezone & Other tab. If the default format is changed, the date formats must comply with the requirements of the Java SimpleDateFormat class:

<http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html>

- ▶ LastUpdateDateTime can be left out in the CSV files.

Before creating a new data source configuration, the required CSV files must be placed in the file system of the application server and accessible by the IBM WebSphere Application Server processes.

In the tutorial *Creating a CSV data source* at <http://youtu.be/4fnnr7w-hDE>, the CSV files are placed in the default directory, but this directory can be different for different data source definitions. A mounted network file system directory can also be used.

Example 3-2 shows a well-formed CSV file.

Example 3-2 Well-formed CSV file

```
ID,STARTDATETIME,ENDDATETIME,NAME,DESCRIPTION,SENDER,SEVERITY,LASTUPDATEDATETIME,TIMEZONEOFFSET,LOCATION
1,2013-11-22 15:30:00.000,2013-11-22 16:30:00.000,Man fall down,Man fall down in the middle of the crowd,robert@example.com,Extreme,2013-11-22 15:30:00.000,0,POINT(-93.256162 44.973056)
2,2013-11-22 15:37:00.000,2013-11-22 16:37:00.000,Car accident,Two cars blocked exit from parking,ann@foo.com,Severe,2013-11-22 15:37:00.000,0,POINT(-93.256267 44.972385)
```

Creating CSV data sources

Follow these steps to create the CSV data sources using the sample files provided with this book:

1. Transfer the sample files `CrowdGathering.csv` and `FirstAidEvents.csv` included in the `sample_data_for_datasources.zip` file to `/opt/IBM/ioc/csv/` on the IBM Intelligent Operations Center application server.
2. Follow the instructions in the video tutorial *Creating a CSV data source* at <http://youtu.be/4fnnr7w-hDE> to create the CSV data source.

3.2.5 CAP data source

Common Alerting Protocol (CAP) is a standard created by the Organization for the Advancement of Structured Information Standards (OASIS) open standards organization. It is an XML-based standard of messages for exchanging emergency or non-emergency alerts and public warnings over all kinds of networks. It is widely used by government agencies in United States, Australia, Canada, and other countries. The standard specification is available on the OASIS website:

<http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>

IBM Intelligent Operations Center supports alert, cancel, and update CAP messages for both geographic information system (GIS) and location maps. The data model for this type of data source supports all properties defined by the standard and it can also store custom properties from the CAP message.

The custom properties can be accessed through the REST API and in the user interface. However, routing expressions can be applied only to the properties defined by the standard.

There are two ways to send CAP messages into IBM Intelligent Operations Center:

- ▶ IBM WebSphere MQ message queue
- ▶ IBM WebSphere Application Server Java Message Service (JMS) resources

Note: To publish CAP messages to the WebSphere MQ message queue, use the following configuration settings:

- ▶ Queue manager: IOC.MB.QM
- ▶ Queue manager port number: 1414
- ▶ Queue manager host name: IBM Intelligent Operations Center analytics server *host name*
- ▶ Queue name: CAP.IN.Q

To publish CAP messages to the WebSphere Application Server JMS resources, use the following configuration settings:

- ▶ Queue Connection Factory:
 - Name: ioc.mb.con.factory
 - JNDI Name: jms/ioc.mb.con.factory
- ▶ Queue:
 - Name: cap.in.q
 - JNDI Name: jms/cap.in.q

After a message arrives to the WebSphere MQ queue or JMS topic in WebSphere Application Server, the CAP MDB in the IBM WebSphere Application Server reads it and stores the message in the target table in the system database.

After the message is stored in the target table, the routing expressions are applied to the store messages and messages are routed to the subsystems of IBM Intelligent Operations Center.

Note: By default, all data items are routed inside IBM Intelligent Operations Center in the internal XML-based format. But there is an exception for CAP messages, which also can be routed in CAP standard format. This option can be useful if it is required to support backward compatibility for custom KPI models developed for version 1.5. To change the default format to CAP for routing messages, navigate to **Administration** → **Solution Administration** → **System Administration** → **System Properties** and change the system property `RouteCAPXML` value to **true**.

Creating Common Alerting Protocol data sources

There are two options to create a CAP data source:

- ▶ Common Alerting Protocol for geospatial maps
- ▶ Common Alerting Protocol for location maps

The difference between these options is in the way that the location data from the data items is interpreted. For more information about using maps in IBM Intelligent Operations Center, see Chapter 7, “Visualizing information with maps” on page 269.

The process of creating a CAP data source is shown in the video tutorial *Creating a CAP data source* at <http://youtu.be/00mQ8phuSgo>.

The tutorial shows how to send a CAP message into the WebSphere MQ queue using the IBM WebSphere MQ Explorer.

To start this tool, log in to the IBM Intelligent Operations Center analytics server graphical user interface with the root user account and run the following commands:

```
xhost + <ioc_ana>  
su - mqm  
/opt/mqm/bin/strmqcfcfg &
```

The source file for the CAP message is available with the samples provided with this book. The file name is `emergency_unit_alert.xml` and it is located in the `sample_data_for_datasources.zip` file.

3.2.6 User interface entry data source

The types of data sources listed in the previous sections can also receive data items entered by the user from the user interface and via API calls.

The user interface entry data source can be extremely useful when there is no data stored in a staging database, CSV file, or represented as CAP message, but there is still a need to receive data items created by the user or through APIs. Creation of this type of data source is simple. See *Creating a data source* at this website:

<http://bit.ly/1nn5Fxa>

3.3 Using APIs to send data into the system

This section includes the following topics:

- ▶ API overview
- ▶ Example how the API is used in standard user interface

3.3.1 API overview

There are three services in IBM Intelligent Operations Center that allow developers to manipulate data sources and send data into the system:

- ▶ **Data source service:** Get, create, update, and delete data sources. This service is used by the data source configuration tool to manage the data source configuration. It is also used in the Operations view to list all the data sources in the filter panes.
- ▶ **Data receiver service:** Get, create, update, and delete data receivers. The data receiver gets data from the source of the data and imports it into the system database. The data receiver is managed automatically using this service for each data source; developers do not need to use this service.
- ▶ **Data injection service:** Add, update, and cancel data items for a data source. Using this service, developers can inject data into any data source that is configured by the system administrator. The data format must conform with the data source configuration and the request must be performed by a user with read-write access to the data source.

All these services have both REST and Java API representations. The REST API is described in detail in the *IBM Intelligent Operations Center V1.6 REST APIs* document, which can be downloaded from <http://www-01.ibm.com/support/docview.wss?uid=swg27039957>.

The Java API documentation is available in the IBM Intelligent Operations Center Information Center at this website:

<http://pic.dhe.ibm.com/infocenter/cities/v1r6m0/topic/com.ibm.ioc.doc/javadoc/overview-summary.html>

Note: Before using any of the REST services described in this section, the user needs to be authenticated to the system with HTTP Basic authentication. When the server provides authentication cookies `LtpaToken2` and `JSESSIONID`, the `com.ibm.ioc.sessionid` parameter must be added to the HTTP header; it must contain the value of the `JSESSIONID` cookie.

For example, to get the list of all the data sources configured in the system, the following HTTP request must be sent to the server:

```
GET https://iop20ihs.ibmplatform.com/ibm/ioc/api/datasource-service/datasources
HTTP/1.1
```

```
Accept: application/json
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Connection: keep-alive
```

Cookie:

```
LtpaToken2=J/5Y5xIRaVNB1+Nvw+TKs9ZXutkgKSRSmqnGZzGg9tr6IqDC/DBRYXWtEdKt6a57J1Sa
S6IHSvQ9L01HfRqNP2m8PDAqIsLjMvA0wghjKe/ytZTWQn5u1Hc+K+Yi2wexVo3z3dm52yDPFoAepQp
i5p7hDHDUr1kCiHVI1e9fVrfSEHy2p4Lb769e25dDbKP+2WrX2YYZsKE5Sau0wR5nGkcRAebtgdCqRy
T8K8izrrC8i3DfF+JyTXp1RfI1HRgpJJzqilcLSBYvdQWk5/6dF7vRMD3kLZa1/97YxcJjiDNmd7z64
wUyw10bttoEEBqi9410hzuyBbXkeUSL66qC1b1b0xBr0Vc6+R0EphKNBvAXXBMKcyABJ8pnMpgNh3E
Y1atYMsuJd+oiqWbBdDXn6gLxJP/rjWxI8+HqeZgvwoCSya9hxypjqLrXgIKnwfqPRGqE6M+9rPTuUE
DZ1hk14KbAMQJ4cvSfX3Ey2kAyqkNfPrjboihRVnGV1taxIrTDSuGoJn3qNBC1ycbj6x/qWFcabq+QW
o1lgK8fXfksRktdDszqH+kG1IKUGVxuBMIt/WiFtQgb7D1A6FvkDnHK6WkhzC/oFyphov+eoxvd1uRR
X/sxiQ1Uaubgj5az0wS3MS1SBGgeqBeBrSJqBGE90r86THypqSY/KxgwMcIhH71+j8H9brbXKBPPRnT
g9MQ; JSESSIONID=0000Nn55ZmH5g8-CCjHZBJz-hiC:-1
```

```
Host: iop20ihs.ibmplatform.com
Referer: https://iop20ihs.ibmplatform.com/
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/30.0.1599.101 Safari/537.36
X-Requested-With: XMLHttpRequest
com.ibm.ioc.sessionid: 0000Nn55ZmH5g8-CCjHZBJz-hiC:-1
```

This approach prevents Cross-Site Request Forgery (CSRF) attacks using a standard REST API.

Code examples of how to use the data source and data injection services are provided in Chapter 8, “Developing mobile applications” on page 299.

3.3.2 Sending data items using the API

One of the best examples of using the REST API is the user interface in IBM Intelligent Operations Center. It is based on HTML 5 and Dojo. Therefore, it is easy to trace the REST API calls using tools for web debugging in modern browsers.

When operators access the Operations view in Citywide → Supervisor: Operations, Citywide → Operator: Operations, or Citywide → Executive: Operations, they can add, update, or cancel data items using the More Actions menu. These actions trigger the REST API call to the data injection service and send HTTP POST, PUT, and DELETE requests that add, update, or delete the data item respectively.

First, the request goes to the REST service. It passes the data item parameters in the Uniform Resource Identifiers (URIs) and data item properties in the body of the HTTP request in the JavaScript Object Notation (JSON) format (see <http://www.json.org/>).

To cancel a data item, only its URI must be specified. Example 3-3 shows a request to delete the data item with id=39 from data source id=14. Both IDs are passed as a part of URI.

Example 3-3 DELETE request to cancel data item with id=39 in the data source with id=14

```
DELETE
https://iop20ihs.ibmplatform.com/ibm/ioc/api/data-injection-service/datablocks/14/
dataitems/39 HTTP/1.1
```

If the data item is deleted successfully, the server returns the HTTP code 204 with no content.

When the operator chooses to add or update a data item, the AJAX script in the browser sends a POST or PUT request with the data item properties in the body of the HTTP request. Example 3-4 shows the call that creates new data item in the data source with id=14.

Example 3-4 REST call that will create a new data item in data source with id=14

```
POST
https://iop20ihs.ibmplatform.com/ibm/ioc/api/data-injection-service/datablocks/14/
dataitems HTTP/1.1

Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US, en; q=0.8
com.ibm.ioc.sessionid: 0000VDGRbQTUIZnkp0mACNS44XH: -1
Connection: keep-alive
Content-Length: 372
Content-Type: application/json
Cookie: LtpaToken2=K2Vl mw0exSRzqEeohPxjX1MINDMqkXVV5NkFK0SDMRjFdBu4URuYmHich0CV1bzc
5SBdyYW01+5saBTiGF/ML7+yhkZzTmx8V9rVgYxY1rMtXF5PQ/fm9ioz6PM1E2rU61YXDNArz16mCKIgtH
7c41uwpM61WGSjZ8Lo9z8fmMD1Usa2M2qGoeRzbl0/5YCjsIgL5Zgv9ZecW8PKAHi9rhB/n9UUYHhV10B
mtKZExR2mGBnxLyDNck+hiS6tChPSEvhFPx8eI6G1DU4RNko7sYW2F0s9xJIeHr1HkVnx5fR0glt6m2nkD
n576DwCNfi9oorCIMzpuICjMIUn16b7Nd+CuP+2HSfp7k6wrV/kqDspJRmhwWOMrOAGAZTBQ2fmPmcQP
Ez5miqJSoXt9NGfv71WIWf6bNdVv5TbIHUTodhhFdSgv10aDtt2Mej9VXFvJsnCsBiMp+yiETStVkaAnLm
h0LN/dbfZhrItUXx34YRqbpFLbq9oZqrRjcyZLfXoKNt7oorrawexjb0bsRZBij/QHuJQHV2rIbX4G0pN
KBwoZ0tzEYCR8vzsK/1LyKhJG1FDfag0Yn0sXAZ1eJm0Db3UTHWxAL2A7sT6KEVBiympJCI4SZTr+hP+z
GgwtJV/VYiNdhfSbKPFJEN4NYdxElFLTvjR6I19psF1eFgy1BaX6cRwj8YGAXwrSj8;
JSESSIONID=0000VDGRbQTUIZnkp0mACNS44XH: -1
Host: iop20ihs.ibmplatform.com
Origin: https://iop20ihs.ibmplatform.com
Referer: https://iop20ihs.ibmplatform.com/wps/porta1!/ut/p/a1/hY_NDoIwEISfxQNXu_Ij4
K0CEQiIiTfCLwYSBAzSp1R8fSvBm-IeNtnJN51ZRfCKsJcPTZWLhnZ5-77J-nIA2_MDC8LE0a0g2CfeUcM
6AJgSyCQAPwD6Hd22fNSAq6pULgbn3XtGOA2Jj8M8Cf_DMisxGB0QEzFUNeqpYW47sZ7grNqhDh5bXkJ
V8-uJRRiVi_UUCBhjIVmrpX4MnkYpSL776a9gK1Hxyx-ymFm9E0EV68AE46XgA!/d15/d5/L2dBISEvZ0F
BIS9nQSEh/
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/30.0.1599.101 Safari/537.36
```

User-TimeZone:240
SessionID: 0000VDGRbQTUIZnkp0mACNS44XH:-1

```
{"NAME":"City Marathon",  
"STARTDATETIME":"2013-10-09T11:45:00-04:00",  
"ENDDATETIME":"2013-10-11T19:30:00-04:00",  
"LOCATION":"Point(-93.25997829437111 44.97333727355225)",  
"Property_6":"11.45",  
"Property_2":"19.30",  
"Property_4":"City Marathon",  
"Property_3":"12/02/2013",  
"Property_7":"0",  
"Property_1":"12",  
"Property_5":"undetermened",  
"DELETEFLAG":false,  
"TIMEZONEOFFSET":"-04:00"}
```

The system returns the HTTP code 200 if the request completed successfully.

From the REST API request, this call is transformed into a Java API call as illustrated in Example 3-5.

Example 3-5 Creating a new data item in data source with id=14 with Java API

```
// [...]  
try {  
    //Getting security context from portlet container  
    SecurityContext sc;  
    int datasourceid = 14;  
  
    Map <String, Object> data = new HashMap<String, Object>();  
    data.put("NAME", "City Marathon");  
    data.put("STARTDATETIME", "2013-10-09T11:45:00-04:00");  
    data.put("ENDDATETIME", "2013-10-11T19:30:00-04:00");  
    data.put("LOCATION", "Point(-93.25997829437111 44.97333727355225");  
    data.put("Property_6", "11.45");  
    data.put("Property_2", "19.30");  
    data.put("Property_4", "City Marathon");  
    data.put("Property_3", "12/02/2013");  
    data.put("Property_7", "0");  
    data.put("Property_1", "12");  
    data.put("Property_5", "undetermened");  
    data.put("DELETEFLAG", false);  
    data.put("TIMEZONEOFFSET", "-04:00");  
  
    // Sending new data item to the the data source  
    DataReceiver dr = (DataReceiver)  
InjectionDataService.singleton().create(uLocale, sc.getUserPrincipal(), null,  
datasourceid, data);  
  
    Map<String, Object> result = dr.getDataItem();  
    } catch(ForbiddenException e) {  
        // TODO: Handle Exception  
    } catch(BadRequestException e) {  
        // TODO: Handle Exception  
    } catch(ConflictException e) {
```

```

        // TODO: Handle Exception
    } catch(Exception e) {
        // TODO: Handle Exception
    }
}
// [...]

```

After the data from the request comes into the system, it goes through the regular flow described in 3.1.1, “Data import process overview” on page 46. All the filters and routing expressions are applied to the data items in the same way as for those that are coming from other sources of data. The only exception is the DELETE request, which is converted into an SQL statement and executed directly to the system database. Figure 3-4 shows the data item process flow when the REST API is called from the user interface More Actions → New Item menu option.

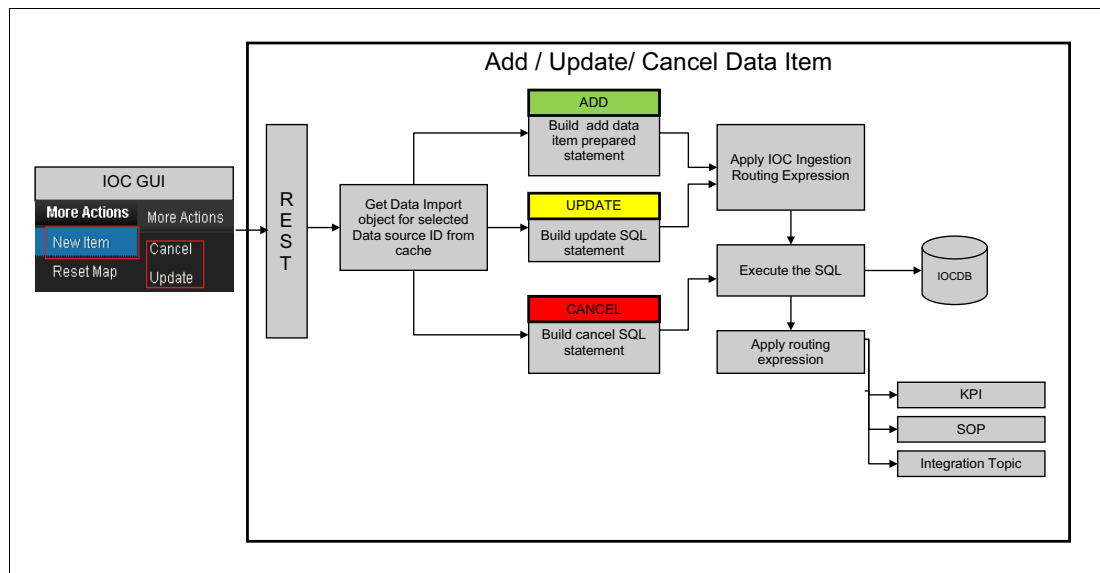


Figure 3-4 Data item processing diagram when the REST API is called from the user interface

3.4 Extension points

There are two extension points that allow you to bring custom data into the system:

- ▶ External service URL: This extension point allows you to show data from external systems or third-party applications in the IBM Intelligent Operations Center user interface with the data items coming from the data sources.
- ▶ Extension properties: This extension point allows you to store more complex data in special properties in the data items from the data sources.

The following sections describe both extension points in more detail.

3.4.1 External service URL data sources

In the standard import process, data items are passed to the data receiver, which performs the routing logic and inserts the data items into the target tables. For example, when the user interface displays information about the data items, it queries the spatial service, which extracts the information out of the target table.

Developers can create an external service URL data source to bypass using the data receiver and target tables to store the data. In this case, to get information about the data items, the request must be sent to the external REST service instead of calling the spatial service and system target tables.

There are three considerations to remember when using this extension point:

- ▶ Because IBM Intelligent Operations Center is not processing the data items, the routing expressions cannot be applied.
- ▶ The external REST service is expected to conform to the system's data and API standard. For example, if the user sends a delete request and the service does not honor it, the record will not be removed from the UI.
- ▶ This type of data source can be created only programmatically; an example is provided in 9.5, "Using an external service URL" on page 335.

3.4.2 Extension properties in the resource definition

The data source resource includes a property called `extensions` where developers can add custom property definitions to a data source if the standard properties are not sufficient for custom applications developed on the IBM Intelligent Operations Center solution. For example, this approach is used to store additional attributes for IBM Intelligent City Planning and Operations.

As with external service URLs, the extension properties can be created or modified only programmatically. Extension properties also require custom JavaScript code to be created on the client side to parse to read and unpack all these properties, which are a list of JSON objects. For more information about customizing the user interface, see Chapter 6, "Customizing the operations display" on page 191.

Note: It is highly advisable to try to map the data to the existing properties before creating new extension properties specific to the custom application.

Managing extension properties using APIs

Extension properties can be managed with the standard data source service through both Java and REST APIs.

The service property `extensions` store a list of JSON objects that represent the data source extension properties and must be created, updated, or deleted as a single object.

Example 3-6 on page 64 shows the `extensions` property with one JSON object for the extension property named `Custom`. It contains the minimal information that needs to be provided to create it. The full list of extension property object properties is available in the *IBM Intelligent Operations Center v1.6 REST APIs* document:

<http://www-01.ibm.com/support/docview.wss?uid=swg27039957>

Example 3-6 Defining a JSON object for an extension property when creating a data source

```
...
"extensions": [...,
{
// extension property name is Custom
"name": "Custom",
// extension property type is CUSTOM (other types are: ACTION or VIEW, which are
// used to create custom data source actions or custom preview cards)
"type": "CUSTOM",
// extension property unique namespace to differentiate it from other applications
// custom properties
"module": "custom.citya.gov",
// String with extension property value
"associated": "<String representation of the custom property value>"
},
...],
...
```

3.5 Troubleshooting data sources

Detailed information to isolate and resolve problems with IBM Intelligent Operations Center are in the topic *Troubleshooting and support* at this website:

<http://bit.ly/11E4PIv>

This section covers additional tips and techniques to help diagnose problems and resolve issues with data sources.

3.5.1 Data items do not reach data source target table in the system database

If the data items are not displayed in the UI, check whether they are reaching the target table in the system database.

Each time that a new data source is created, a new target table for the data source is created in the IOCDDB system database to store the data items for the data source. While data is imported into the system, the data items appear in the target tables.

The format of the target table name is `IOC.TARGET_TABLE_<ID>` where `<ID>` is the unique identifier of the data source. System administrators can use the following steps to identify the unique ID of the data source:

1. Authenticate to the system with the credentials of the user who created the data source.
2. Execute a REST call to list all data sources registered in the system (see the example in the shaded note box in 3.3.1, “API overview” on page 58).
3. Parse the JSON object received in the server’s response to extract the data source name or description. Example 3-7 on page 65 shows a server reply.

Example 3-7 Data source properties example from the server reply

```
{ "id": 26,
  "lastUpdateDate": 1376493682285,
  "messages": [],
  "type": "CSV",
  "label":
  "{ \"group\": \"DataSourceI18n\", \"key\": \"i18n_4ac9f374-fc5d-4d9d-8d02-6a1826e9301a
  \"}",
  "i18nLabel": "MyDataSourceName",
  "description":
  "{ \"group\": \"DataSourceI18n\", \"key\": \"i18n_5a3520f4-c8bb-49fc-820e-7ea31c406673
  \"}",
  "i18nDescription": "MyDataSource Description",
  [...]
}
```

Note: The format of the target table name might change in the future. Monitor the technotes released for IBM Intelligent Operations Center on the IBM Support Portal: http://www-947.ibm.com/support/entry/portal/product/smarter_cities/ibm_intelligent_operations_center?productContext=-1247937804

After the target table name for the data source is identified, use a DB2 client application, such as IBM Data Studio or DB2 command line processor, to check whether the data items are stored in the table.

Follow these steps with IBM Data Studio:

1. Log in to the IBM Intelligent Operations Center data server graphical user interface with the root user account.
2. Open a terminal session and run the following commands:

```
xhost + <ioc_db>
su - db2inst2
/opt/IBM/DS3.2.0/eclipse &
```
3. Right-click **All Databases localhost** → **db2inst2** → **IOCDB**, click **Connect**, and enter the password for **db2inst2**.
4. Click **Tables**. Find the data source target table by the name identified in Step 3 on page 64.
5. Double-click the table name to open it and check whether there is any data in the table.

If the data source is configured and created successfully, but the data items do not appear in the target database table, enter the command shown in Example 3-8 to enable portal server tracing in the WebSphere Application Server 8.0 Deployment Manager console. Then, attempt to recreate the issue; for example, create another data source.

For detailed information about the procedure, see *Enabling tracing and viewing logs on WebSphere Portal*:

<http://bit.ly/1jzRs9J>

Example 3-8 Enabling logging to debug data import problems

```
*=warning: com.ibm.ioc.logger.DataReceiverLogger=all
```

The typical problems can be hidden in the source of the data. The following list shows issues that can prevent data items from getting into the database table:

- ▶ Common issues for DB2 database tables data source:
 - To support updates of data items from a staging table, the `LastUpdateDateTime` column must be present and valid. That is, the `LastUpdateDateTime` field must be populated if present. If there is no `LastUpdateDateTime` column, there is no way to check for updates so updates will not take place.
 - The `Location` field must be either in valid WKT format or valid separate latitude and longitude coordinates as shown in Example 3-1 on page 55.
- ▶ Common issues for CSV files data source:
 - Check that the CSV file meets all requirements listed in 3.2.4, “CSV data source” on page 55.
 - The system administrator might choose to change the data type of data source parameters when configuring the Key & Full Properties step in the data source configuration tool. In this case, the type of data for each property in the CSV file must match what is configured by the administrator.

3.5.2 Data steps to be imported after a database failure

If, for example, a data source has a polling interval of one minute and it fails ten times, it will be disabled; this can happen in approximately eight minutes. Example 3-9 shows a `wsadmin` script to restart the scheduler and any disabled tasks.

Example 3-9 Sample script to restart the data receiver after database failure

```
> import sys, time
> sched = AdminControl.queryNames('WebSphere:*,type=WASScheduler,name=Scheduler_ioc.Scheduler')
> AdminControl.invoke(sched, 'stopDaemon')
> time.sleep(5)
> AdminControl.invoke(sched, 'startDaemon', '0')
```

Run this script on the IBM Intelligent Operations Center application server. For instructions about running scripts using the `wsadmin` tool, see *Starting the wsadmin scripting client using wsadmin scripting*:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.base.doc/info/aes/ae/txml_launchscript.html

3.5.3 Things to check if data items are not displayed on the map

If the data items are successfully stored in the target table in the system database and there are no errors in the log, but the data items are still not displayed on the map, check for the following problems:

- ▶ Wrong Date & Time selection in the filter panel in the Operations view. The `StartDateTime` of the data item must fall into the selection in the filter panel. `EndDateTime` is irrelevant for this search.
- ▶ Wrong map positioning. Check the coordinates in the data items. Note that the current map area also applies to the list and the map. If in doubt, switch to the world view by clicking the globe icon as shown on Figure 3-5 on page 67.

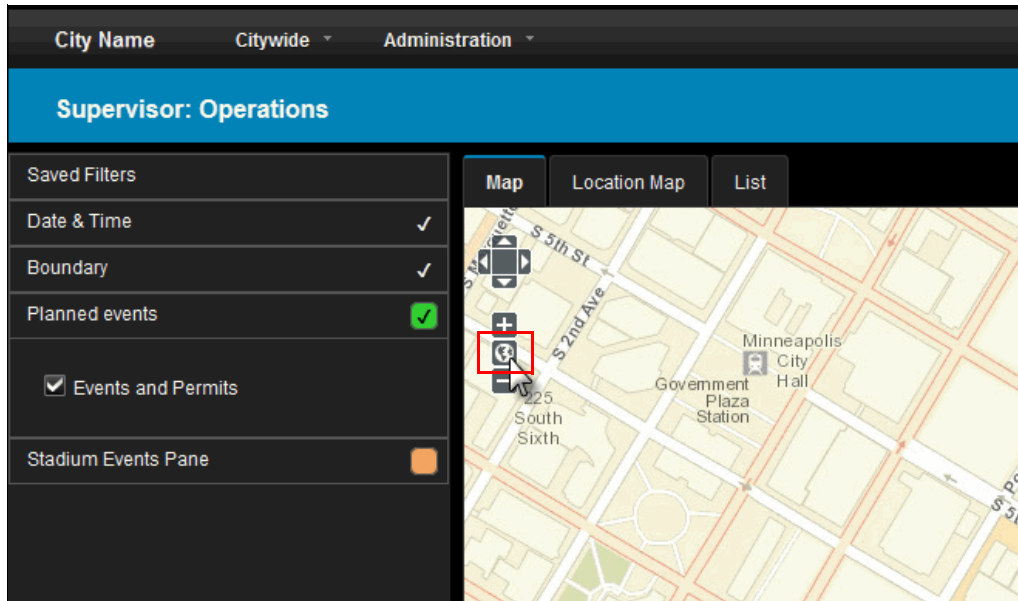


Figure 3-5 Switching to the world view

- ▶ The area name field of the minimal properties is populated during the data source creation. The area name field is used for location maps and, if populated, the data items will not be displayed on the geospatial map. To reset the value from this parameter, re-create the data source.



Tracking key performance indicators

This chapter describes the key performance indicator (KPI) subsystem of IBM Intelligent Operations Center. This chapter provides an overview of the KPI-related terminology and capabilities provided by IBM Intelligent Operations Center. This chapter includes examples of how to create and manage KPIs. Guidelines and tips for implementation and problem determination are also provided.

This chapter includes the following topics:

- ▶ 4.1, “Overview” on page 70
- ▶ 4.2, “Use case overview” on page 73
- ▶ 4.3, “Setting up the development environment” on page 74
- ▶ 4.4, “Implementation summary” on page 74
- ▶ 4.5, “Implementation details” on page 74
- ▶ 4.6, “Problem determination” on page 94
- ▶ 4.7, “Programmatic access to KPIs” on page 102
- ▶ 4.8, “Additional considerations” on page 105

4.1 Overview

KPIs are quantifiable measurements of the improvement or deterioration in the performance of an activity critical to the success of a business. A KPI can have a target, a set of ranges, or both; it provides a measurement of how well a business is achieving its objectives. A KPI obtains its value either from aggregating a metric or from a calculation based on other KPIs.

KPIs are quantifiable measurements employed by organizations to monitor and assess performance. While KPIs differ from one organization to the next, their purpose remains the same. They help reduce complex organizational performance information into a consumable format that allows organizations to more easily assess performance by comparing KPIs to the organization's stated objectives.

4.1.1 Terminology

The following KPI-related terms are used in IBM Intelligent Operations Center (see Figure 4-1):

- ▶ **Owning organization:** An owning organization is a container object that can be used to link a KPI model to an icon and a display label. It can contain one or more child KPIs in a hierarchical structure.
- ▶ **KPI legend:** The KPI legend offers an explanatory key to the data being displayed. It gives the user information about the current state and other possible states of the KPIs being displayed.
- ▶ **KPI view filter:** The KPI view filter provides a means of filtering the displayed KPIs based on their status. The view filter is populated based on the available states defined in the KPI legend.
- ▶ **Bread crumb:** Shows the current position in the KPI hierarchy. It also allows a user to navigate to previous nodes in the hierarchy.
- ▶ **KPI configuration tool:** Tooling is provided in the IBM Intelligent Operations Center Solution Administration view to configure KPIs as shown in Figure 4-2 on page 71.

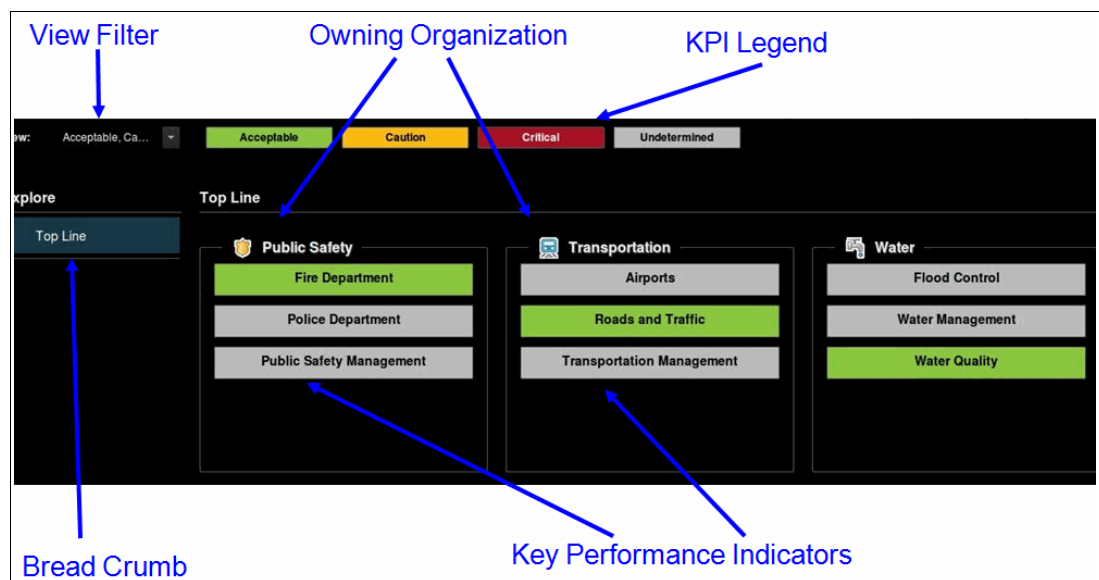


Figure 4-1 KPI terminology and the IBM Intelligent Operations Center user interface

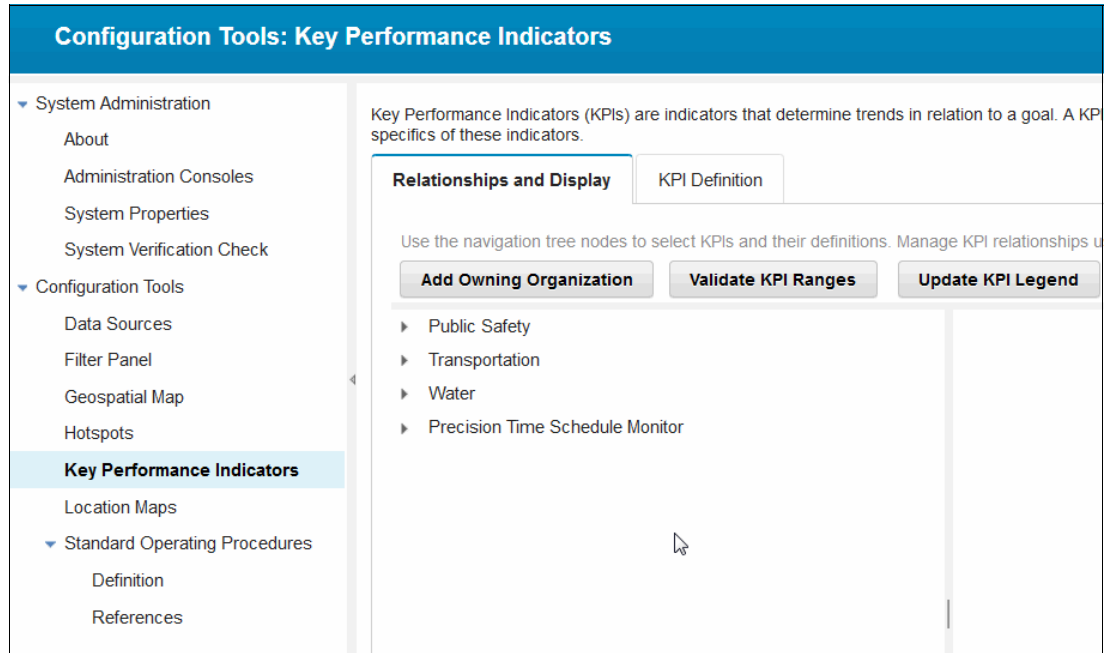


Figure 4-2 KPI configuration tool

4.1.2 IBM Intelligent Operations Center capabilities

Intelligent Operations Center provides a KPI subsystem, tooling, and visual components that allow organizations to define and monitor KPIs that are meaningful to their goals and objectives.

The IBM Intelligent Operations Center KPI subsystem consists of the following components:

- ▶ IBM Business Monitor
- ▶ Sample IBM Intelligent Operations Center IBM Business Monitor models
- ▶ IBM Intelligent Operations Center KPI configuration tool
- ▶ IBM Intelligent Operations Center KPI portlet
- ▶ REST APIs for client-side code to interact with the KPI subsystem
- ▶ Service APIs for back-end services to interact with the KPI subsystem

A set of sample KPIs is installed with the product. The sample KPIs demonstrate the functionality and monitoring capability of the IBM Intelligent Operations Center KPI subsystem and visual components. The source for the sample KPIs is provided to allow organizations to view or modify the KPIs using the KPI tooling. Figure 4-3 on page 72 shows the sample KPIs provided with IBM Intelligent Operations Center.

Water	Transportation	Public Safety
Flood Control Water Levels Flow Discharge City River Water Level City Lake	Airports Delayed Flights Delayed Flights Airport One Delayed Flights Airport Two	Fire Department Firefighter Injuries Firefighter Injuries Fire Station One Firefighter Injuries Fire Station Two
Water Management Strategical Planning Water Leakage Water Supply vs Demand	Roads and Traffic Road Events Severe Traffic Accidents	Police Department Crime Response Time Crime Response Time Precinct One Crime Response Time Precinct Two
Water Quality Physical Indicators Turbidity pH	Transportation Management Revenue Bridges and Tunnel Tolls Parking Facilities Revenue	Public Safety Management Public Safety Budget EMS Department Budget Fire Department Budget Police Department Budget

Figure 4-3 List of KPIs provided in IBM Intelligent Operations Center v1.6

Note: The KPIs in IBM Intelligent Operations Center are driven by the data items that are ingested through the data sources. When the administrators configure the data source, they can create a routing expression to route selected data items to the KPI subsystem. The values of the properties from these data items are used by the monitor model running in IBM Business Monitor to change the value (and corresponding color) of the KPI.

4.1.3 Types of KPIs

Two types of KPIs can be managed in IBM Intelligent Operations Center v1.6:

- ▶ **Aggregate KPI:** Aggregate KPIs are calculated using a metric defined within IBM Business Monitor and an aggregate function, such as average, maximum, minimum, sum, count (number of occurrences), or standard deviation.
- ▶ **Expression KPI:** Expression KPIs can be used to generate results based on other KPIs.

4.1.4 Tooling choices

Both aggregate and expression KPIs can be created by one of the following two methods:

- ▶ Using IBM Business Monitor toolkit: These KPIs are referred to as *modeled KPIs*.
- ▶ Using the IBM Intelligent Operations Center KPI configuration tool: These KPIs are referred to as *dashboard KPIs*.

There are certain restrictions in using the KPI configuration tool based on how the KPIs were created. Table 4-1 on page 73 highlights the restrictions.

Table 4-1 Modeled versus dashboard KPIs

Dashboard KPIs	Modeled KPIs
Created using KPI configuration tool.	Created using IBM Business Monitor toolkit.
Everything but the model associated with the KPI can be changed using the KPI configuration tool.	<p>Only the following items can be changed using the KPI configuration tool:</p> <ul style="list-style-type: none"> ▶ KPI name ▶ KPI description ▶ KPI ranges ▶ KPI targets <p>All other changes can only be made using the IBM Business Monitor toolkit.</p>
KPI can be deleted using the KPI configuration tool.	Deletion is not possible using the KPI configuration tool. The IBM Business Monitor toolkit must be used to delete modeled KPIs.

4.2 Use case overview

This section describes the steps for identifying, defining, testing, and deploying a KPI in IBM Intelligent Operations Center v1.6.

Because there are multiple events happening in City A, the mayor wants to monitor events with large crowd gathering every hour. The assumption for this use case is that any event with crowd size more than 10 is considered large. Table 4-2 shows the thresholds for the KPIs in this use case.

Table 4-2 Crowd size thresholds

Crowd size	Legend	Color
Fewer than 10	Acceptable	Green
Between 10 and 20	Caution	Yellow
20 or more	Severe	Red

4.2.1 Actors

The following actors are in this use case:

- ▶ City mayor
- ▶ Administrator

4.2.2 Desired outcome

The city mayor is able to monitor the total crowd size of large events happening in the city so that the adequate presence of law enforcement and public safety officials is ensured.

4.2.3 IBM Intelligent Operations Center programming artifacts

The following artifacts are used to implement this use case:

- ▶ Monitor model that is defined using the IBM Business Monitor development toolkit
- ▶ KPI definition that is created using the KPI configuration tool

4.3 Setting up the development environment

The development environment for this use case includes the setup of the following components:

- ▶ IBM Rational Application Developer with the IBM Business Monitor development toolkit or IBM Integration Designer. For information about setting up the IBM Business Monitor development toolkit, see *Installing and removing the toolkit*.

<http://bit.ly/1r1N11N>

Note: IBM Business Monitor V8.0.1.1 and IBM Rational Application Developer 8.0.4 are included with the IBM Intelligent Operations Center software as part of the IBM Intelligent Operations Center 1.6 Developer's toolkit. These products can only be used to configure KPIs to be used in the IBM Intelligent Operations Center platform. IBM Integration Designer is not included and must be purchased separately.

- ▶ The sample monitor models are available in the directory `/opt/IBM/ioc/monitor/models` on the IBM Intelligent Operations Center application server. The file name is `ioc_sample_monitor_models.zip` and it is in Project Interchange Format. This sample file can be imported to the workspace by copying it to your local drive and then clicking File → Import → Other → Project Interchange.

4.4 Implementation summary

The process to define KPIs in IBM Intelligent Operations Center is summarized:

1. Identify the input data.
2. Define the monitor model.
3. Create the KPI.
4. Establish the KPI relationships.
5. Visualize the new KPI.

4.5 Implementation details

This section includes detailed information to implement this use case.

4.5.1 Identifying input data

The first step in defining a KPI is identifying the data items that will drive the monitoring model. In this use case, the assumption is that the input data includes an attribute called `CRWDSIZE` that provides the total crowd size in a gathering.

For information about how to create the data source, see 3.2.4, “CSV data source” on page 55. The `CRWDSIZE` attribute is defined as part of the Key & Full Properties tab as shown in Figure 4-4 on page 75.

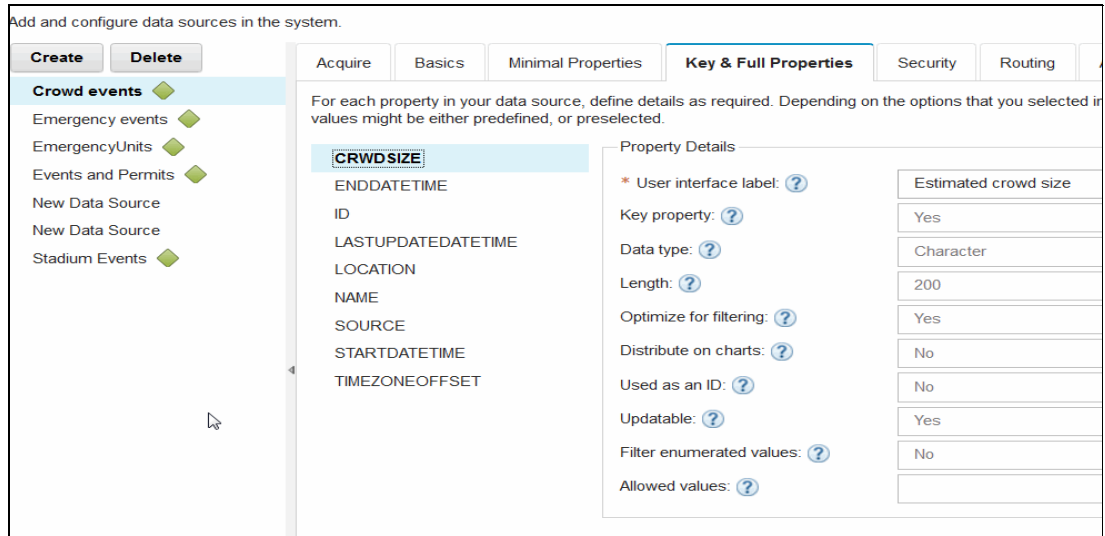


Figure 4-4 CRWDSIZE attribute definition

4.5.2 Defining the monitor model

This section shows how to enhance the sample public safety monitor model to the crowd size model. Then, the KPI configuration tool is used to create a new KPI and add it to the existing Police Department KPI as a child for visualization.

Define the monitor model:

1. Launch IBM Rational Application Developer (with IBM Business Monitor toolkit installed) and import the sample monitor models as Project Interchange (see Figure 4-5).

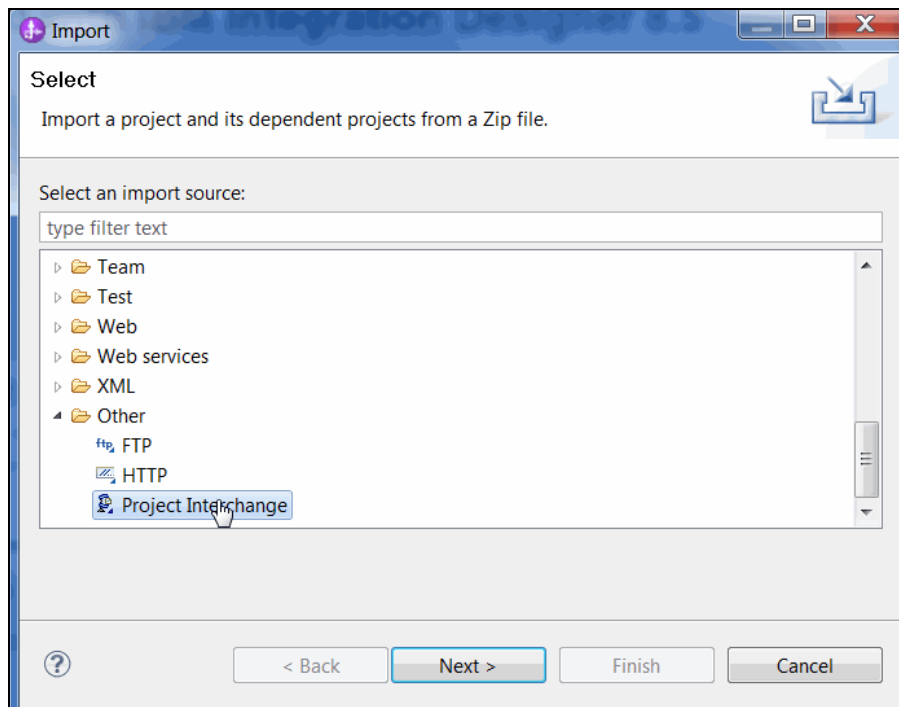


Figure 4-5 Importing sample monitor models as Project Interchange

2. Update the sample routing schema to include the new Crowd Size attribute:
 - a. Select the **IOC-routing-v1.6.0.1.xsd** file as shown in Figure 4-6.

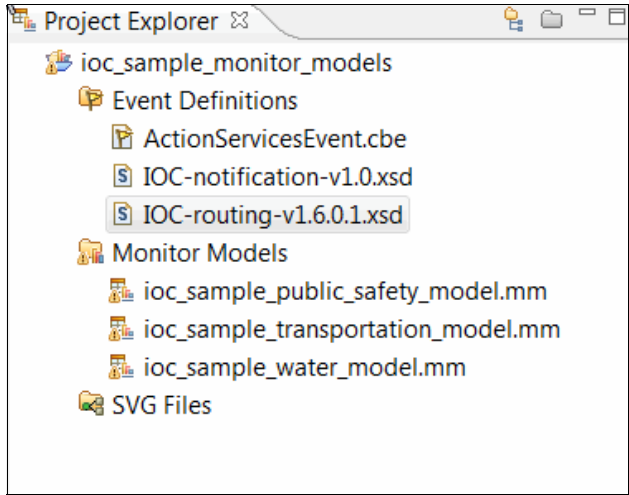


Figure 4-6 Selecting IOC-routing-v1.6.0.1.xsd

Note: For Common Alerting Protocol (CAP) data sources, the system uses the CAP data format instead of the routing schema. For more information, see 3.2.5, “CAP data source” on page 56.

- b. Right-click and select **Open With** → **Text Editor** to launch the schema file in a text editor as shown in Figure 4-7.

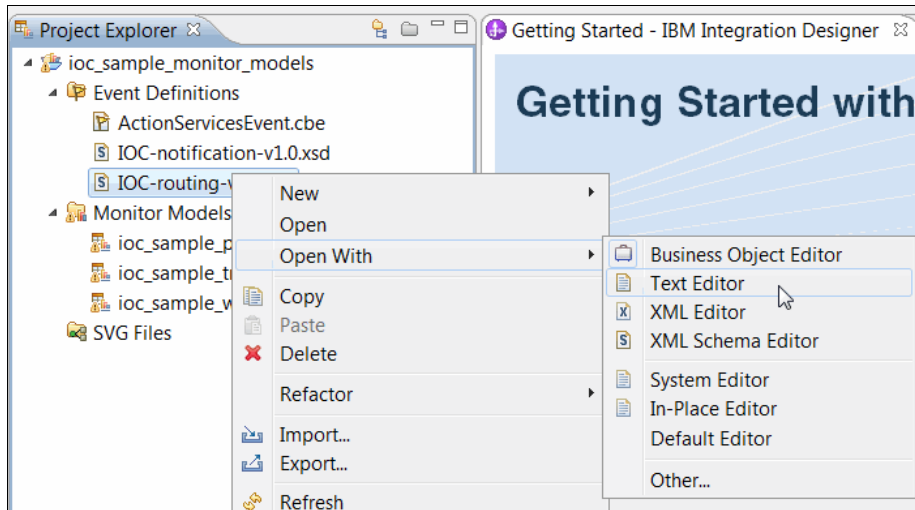


Figure 4-7 Launching the text editor

- c. Add the CRWDSIZE element under the Key/Full properties remarks as shown in Figure 4-8 on page 77.

```

<!--
* Key/Full properties
*/
-->
<element name="CRWDSIZE" type="string" minOccurs="0" maxOccurs="1" />
<element name="SUPPLY" type="string" minOccurs="0" maxOccurs="1" />
<element name="DEMAND" type="string" minOccurs="0" maxOccurs="1" />

```

Figure 4-8 Inserting the CRWDSIZE element

The updated sample routing schema is shown in Example 4-1.

Example 4-1 Modified routing XSD

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
/*
* Licensed Materials - Property of IBM
*
* 5725-D69
*
* (C) Copyright IBM Corp. 2013 All rights reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*
* DISCLAIMER OF WARRANTIES. The following code is sample code
* created by IBM Corporation. This sample code is warranted to perform its
* intended function only if used unmodified. If you modify
* this code then it is considered provided "AS IS", without
* warranty of any kind. Notwithstanding the foregoing, IBM
* shall not be liable for any damages arising out of your use
* of the sample code, even if they have been advised of the
* possibility of such damages.
*/
-->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace = "urn:ibm:industry:solutions:ioc:routing:1.6.0.1"
  xmlns:iocrt = "urn:ibm:industry:solutions:ioc:routing:1.6.0.1"
  xmlns:xs = "http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="qualified">
  <element name="record">
    <annotation>
      <documentation>IOC data source record</documentation>
    </annotation>
    <complexType>
      <all>
        <element name="DATASOURCEID" type="dateTime" minOccurs="1" maxOccurs="1" />
        <element name="UNIQUETEXTIDENTIFIER" type="string" minOccurs="1" maxOccurs="1" />
        <element name="OBJECTID" type="dateTime" minOccurs="1" maxOccurs="1" />
        <element name="STARTDATETIME" type="dateTime" minOccurs="1" maxOccurs="1" />
        <element name="ENDDATETIME" type="dateTime" minOccurs="1" maxOccurs="1" />
        <element name="LASTUPDATETIME" type="dateTime" minOccurs="1" maxOccurs="1" />
        <element name="ASSESSMENTDATETIME" type="dateTime" minOccurs="1" maxOccurs="1" />
        <element name="TIMEZONEOFFSET" type="integer" minOccurs="1" maxOccurs="1" />

```

```

<element name="LOCATION" type="string" minOccurs="1" maxOccurs="1" />
<element name="NAME" type="string" minOccurs="1" maxOccurs="1" />
<element name="AREAS" minOccurs="0" maxOccurs="1">
<complexType>
  <all>
    <!--
      * Area Types
    */
    -->
  </all>
</complexType>
</element>
<element name="KEYANDFULL" minOccurs="0" maxOccurs="1">
<complexType>
  <all>
    <!--
      * Key/Full properties
    */
    -->
    <element name="CRWDSIZE" minOccurs="0" maxOccurs="1">
  </all>
</complexType>
</element>
</all>
</complexType>
</element>
</schema>

```

Note: Interim fix PO02174 provides updates to the IBM Intelligent Operations Center database and to the IBM Intelligent Operations Center V1.6.0.1 routing schema. For more information, see the IBM Intelligent Operations Center database and routing schema updates at this website:

<http://www-01.ibm.com/support/docview.wss?uid=swg24036674>

3. Modify the sample public safety monitor model to include the new monitoring context:
 - a. Select **ioc_sample_public_safety_model.mm** from the Monitor Models list. The Monitor Details Model is displayed on the right pane.
 - b. Right-click **ioc_sample_public_safety_model** and select **New → Monitoring Context** to define a new monitoring context as shown in Figure 4-9 on page 79.

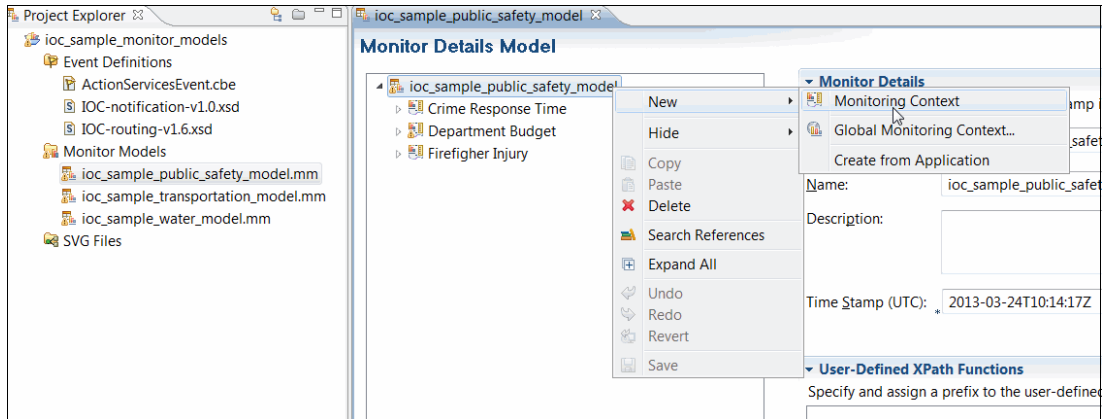


Figure 4-9 Creating a new monitoring context

- c. Enter **Crowd Size** in the Name field. The system automatically assigns the ID based on the name provided as shown in Figure 4-10.

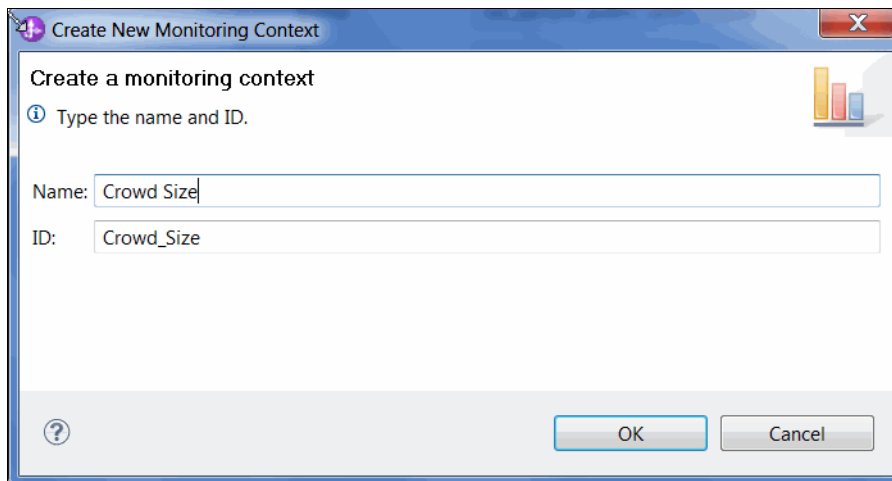


Figure 4-10 Naming the monitoring context Crowd Size

4. Define an inbound event that will trigger the context:
 - a. Right-click the newly created **Crowd Size** monitoring context and click **New** → **Inbound Event** as shown in Figure 4-11.

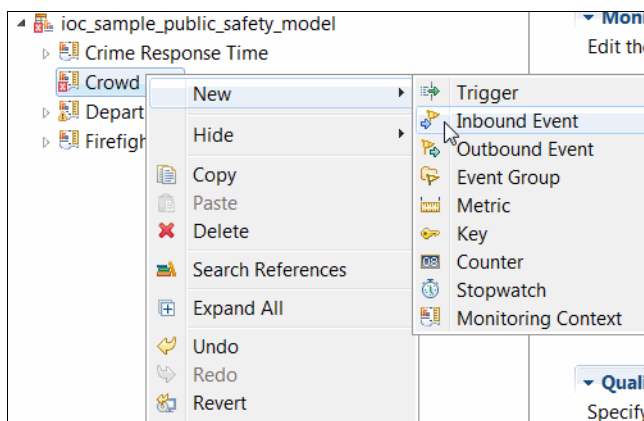


Figure 4-11 Creating an inbound event

- b. Enter **Crowd Size Inbound Event** in the Name field. The system automatically assigns the ID based on the provided name as shown in Figure 4-12.

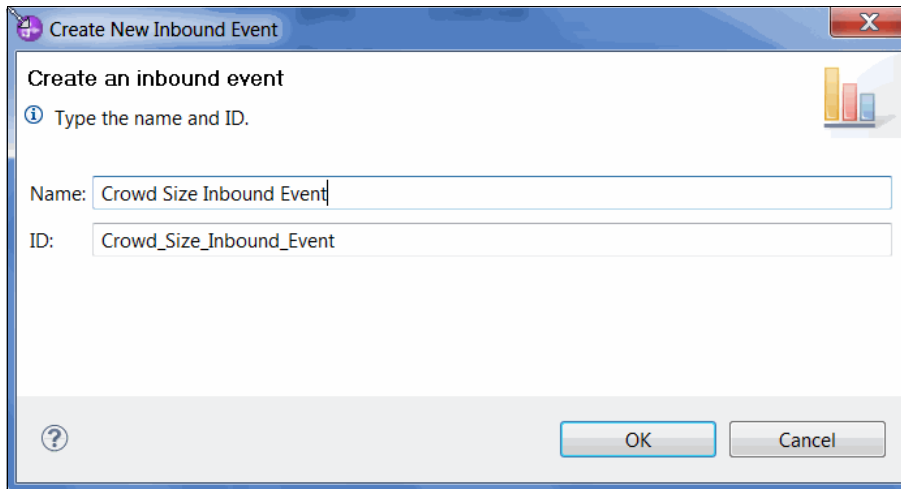


Figure 4-12 Naming the inbound event Crowd Size Inbound Event

5. For the event part data type, select the routing schema updated in step 2 on page 76 as shown in Figure 4-13 on page 81 (the following letters correspond to the letters in Figure 4-13 on page 81):
 - a. Navigate to Event Parts and click **Add**.
 - b. In the Create New Event Part Type window, enter **record** in the Name field. The system automatically generates the ID based on the name.
 - c. Click **Select Type**.
 - d. In the Select Event Part Data Type window, select **Choose the data type from the XML schemas accessible from this monitor option**.
 - e. Drill down and select the **iocrt:record** item.
 - f. Click **Finish**.

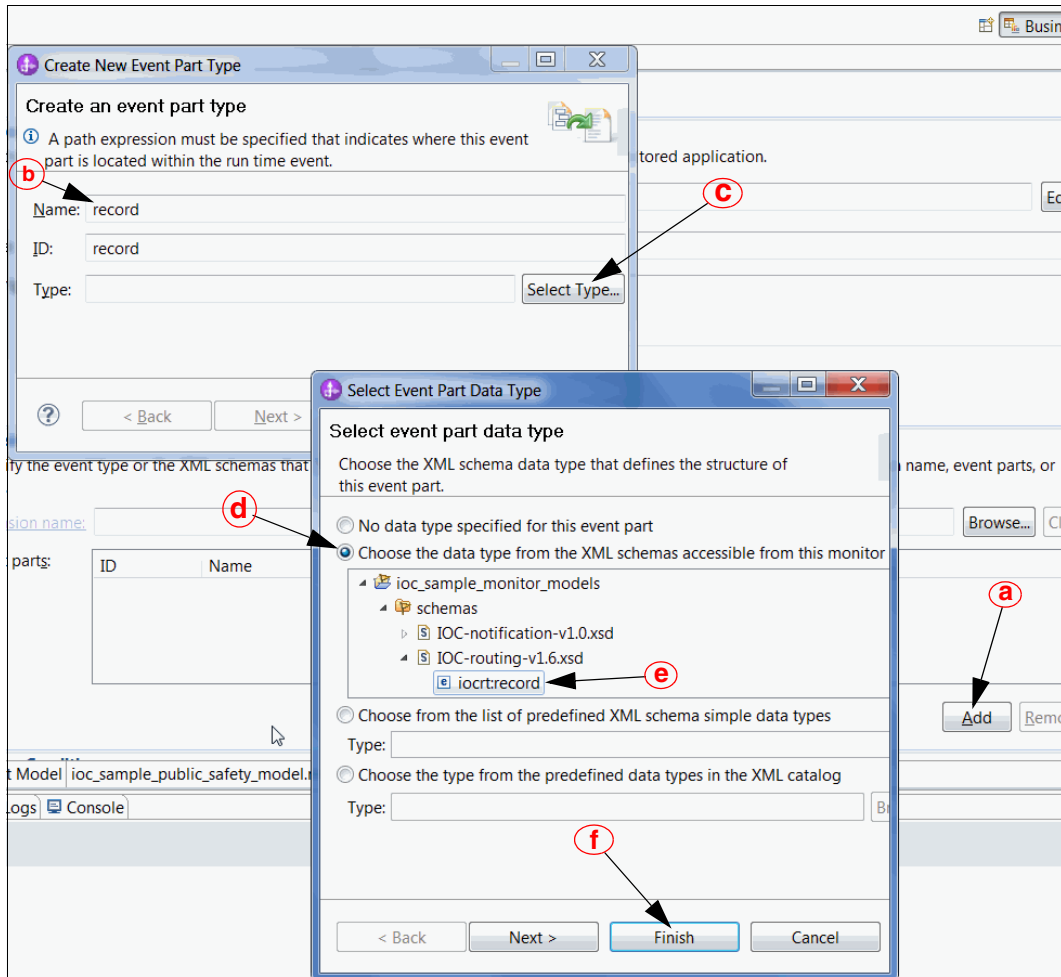


Figure 4-13 Using routing schema for event parts

6. Create a monitoring context key.

The *context key* uniquely identifies the monitoring context instance that you want to monitor. Keys get their values from the inbound event when the monitoring context instance is created. Follow these steps:

- a. Right-click the **Crowd Size** monitoring context and click **New** → **Key** as shown in Figure 4-14 on page 82.

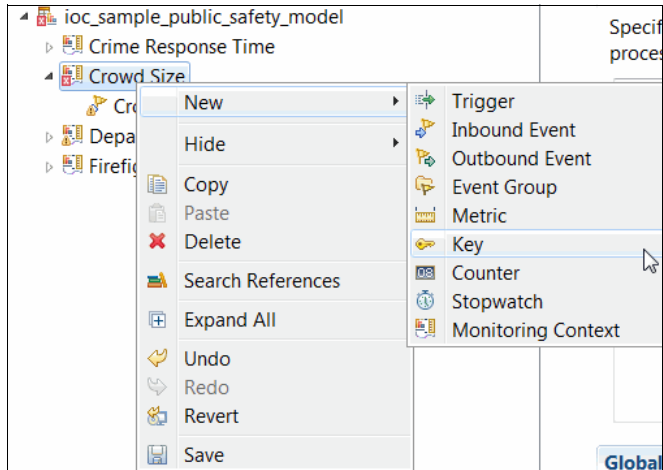


Figure 4-14 Creating a new key

- b. Enter **Crowd Size Key** in the Name field. The system automatically assigns the ID based on the name as shown in Figure 4-15.

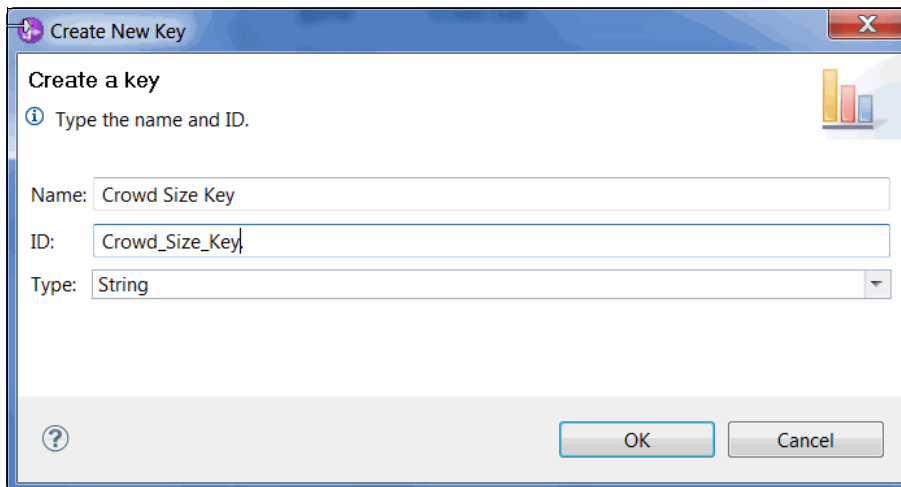


Figure 4-15 Naming the key Crowd Size Key

- 7. Select the INCIDENT_NUMBER attribute as the key value expression as shown in Figure 4-16 on page 83 by following these steps (the following letters correspond to the letters in Figure 4-16 on page 83):
 - a. Navigate to the Key Value Expressions section and click **Add**.
 - b. Press Ctrl+Spacebar on the first row of the expression table to launch context assist.
 - c. Expand the record section and click **INCIDENT_NUMBER**.

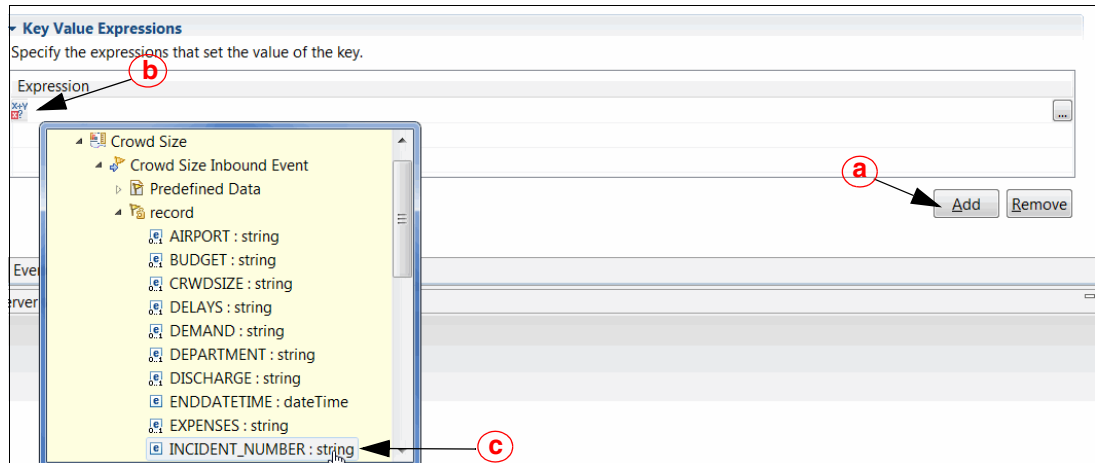


Figure 4-16 Use *INCIDENT_NUMBER* as key value expression

8. Specify an inbound event correlation expression.

The *inbound event correlation expression* tells IBM Business Monitor what monitoring context instance (or instances) will receive the event at runtime. Follow these steps:

- a. Select the **Crowd Size Inbound Event** and navigate to the Correlation Expression section as shown in Figure 4-17.

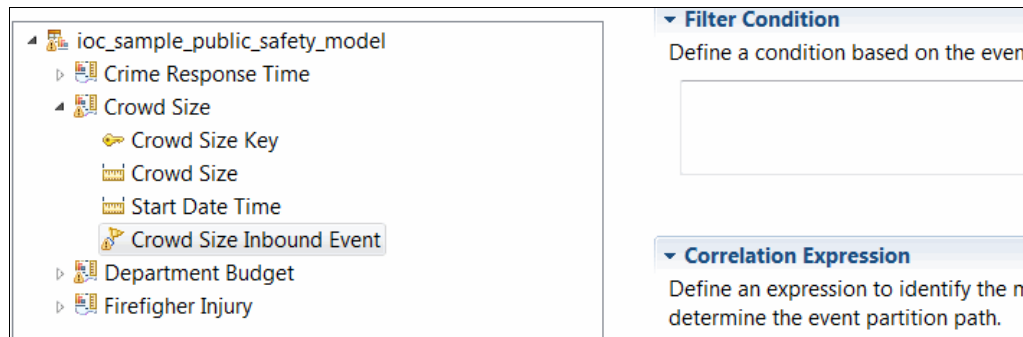


Figure 4-17 Navigating to the Correlation Expression for Crowd Size Inbound Event

- b. Specify the correlation expression (see Figure 4-18 on page 84):

- i. Enter the correlation expression:

`Crowd_Size_Key = Crowd_Size_Inbound_Event/record/iocrt:INCIDENT_NUMBER`

- ii. Verify that **Create new instance** is selected in the “If no instances are found” field.
- iii. Verify that **Treat as error** is selected in the “If one instance is found” field.

Correlation Expression

Define an expression to identify the monitoring context instance or instances that receive the event at run time. determine the event partition path.

Crowd_Size_Key = Crowd_Size_Inbound_Event/record/iocrt:INCIDENT_NUMBER |

If no instances are found

If one instance is found

If multiple instances are found

Figure 4-18 Updating the correlation expression

9. Define the Crowd Size Metric.

A *metric* is a holder for information, typically a business performance measurement in a monitoring context. Follow these steps:

- a. Right-click the **Crowd Size** monitoring context and select **New** → **Metric** as shown in Figure 4-19.

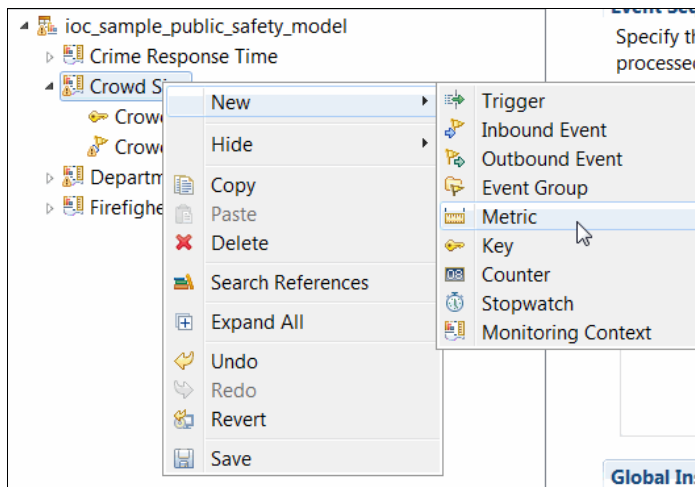


Figure 4-19 Creating a new metric (window 1 of 2)

- b. In the Create New Metric window (see Figure 4-20 on page 85), follow these steps:
 - i. Enter **Crowd Size** in the Name field.
 - ii. The system automatically generates the ID based on the name.
 - iii. Select **Integer** as the Type from the pull-down list.
 - iv. Click **OK**.

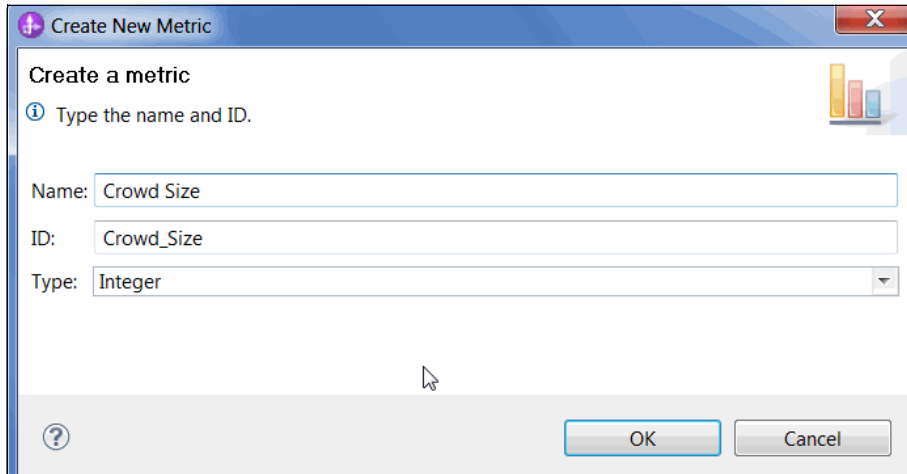


Figure 4-20 Creating a new metric (window 2 of 2)

- c. Specify the CRWDSIZE attribute as the expression to populate the Metric. Navigate to the Metric Value section and click **Add** to add a new expression. In the Expression field, enter `xs:integer(Crowd_Size_Inbound_Event/record/iocrt:CRWDSIZE)`. See Figure 4-21.

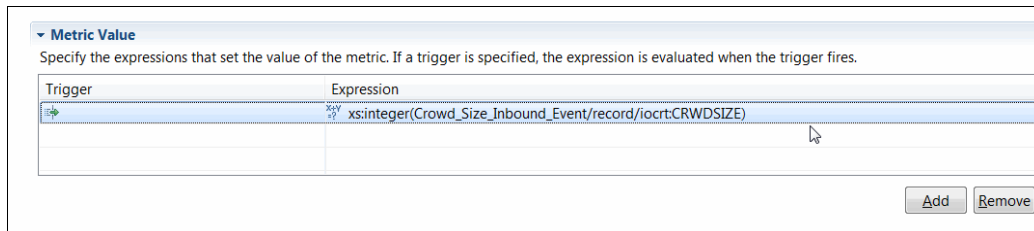


Figure 4-21 Updating the metric expression value

10. Define another metric for the start time:

- a. Right-click the **Crowd Size** monitoring context and click **New** → **Metric** as shown in Figure 4-22.

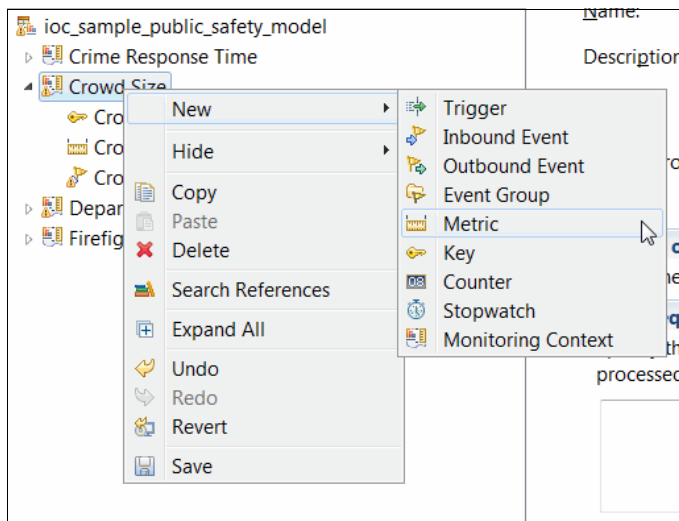


Figure 4-22 Creating a new metric

- b. Enter **Start Date Time** in the Name field and select **DateTime** from the Type pull-down list as shown in Figure 4-23.

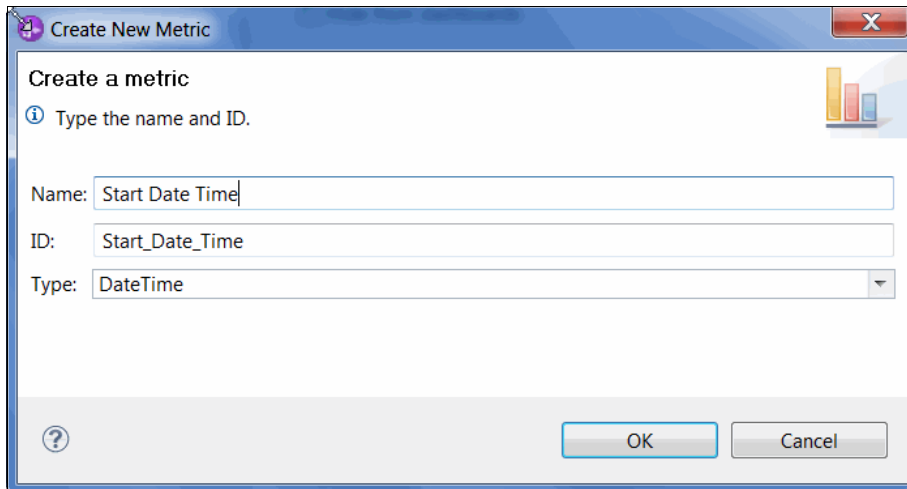


Figure 4-23 Creating the Start Date Time metric

- c. Select the **STARTDATETIME** attribute of the Crowd Size Inbound Event record as the expression for the metric as shown in Figure 4-24.

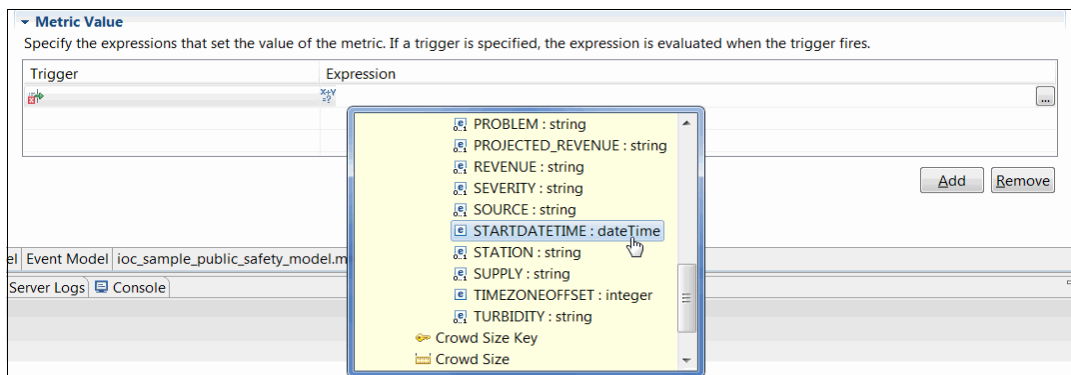


Figure 4-24 Specifying the metric expression

The monitoring context for crowd size is now created.

11. Generate the Monitor Java EE project.

On the left pane, right-click **ioc_sample_public_safety_model.mm** and click **Generate Monitor Java EE Projects** as shown in Figure 4-25 on page 87.

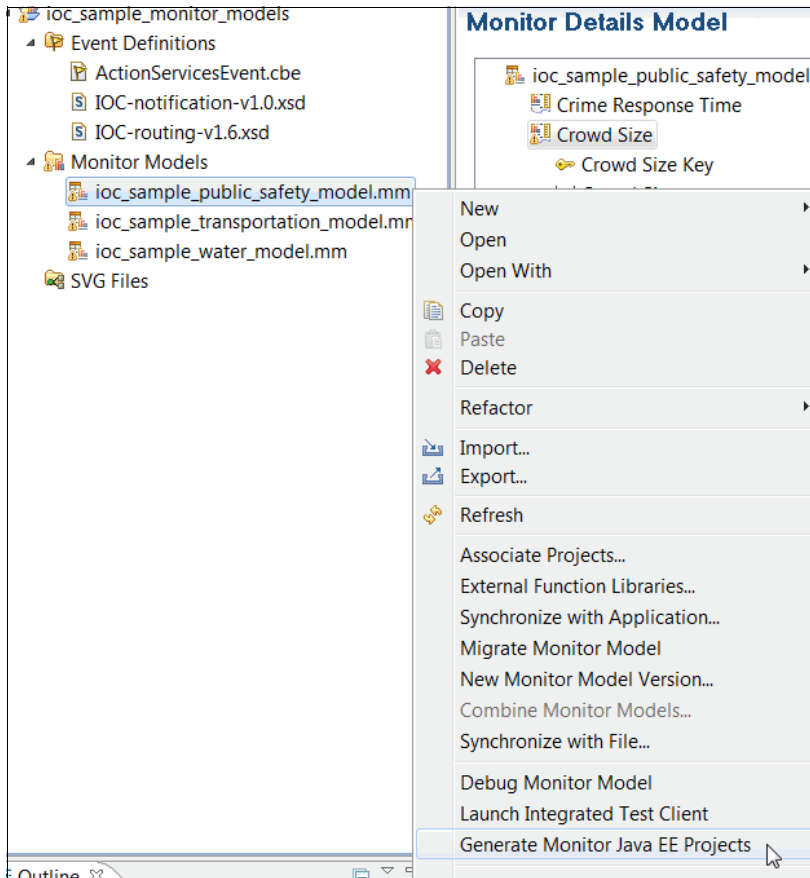


Figure 4-25 Generating the Monitor Java EE Project

12. Switch to the Java EE Perspective.

Right-click **ioc_sample_public_safety_modelApplication** and click **Export** → **EAR file** as shown in Figure 4-26.

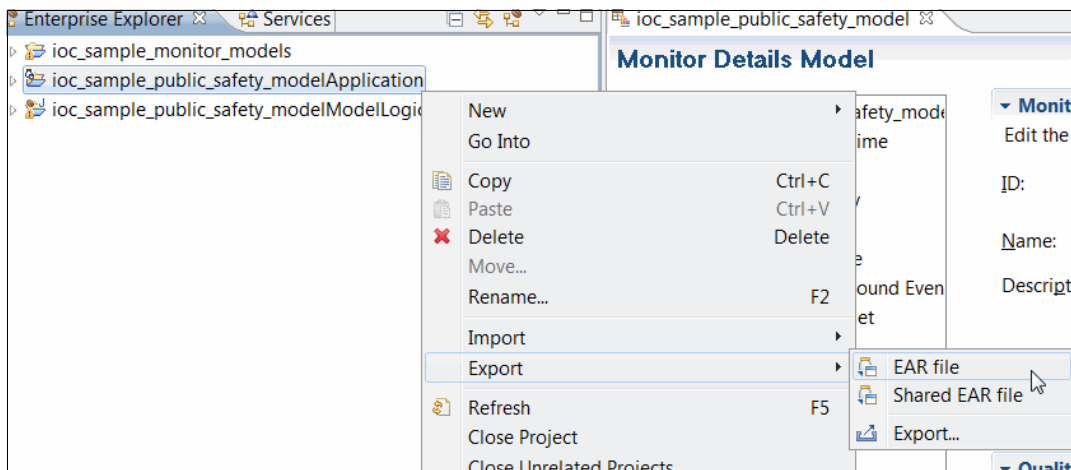


Figure 4-26 Exporting the EAR file

The EAR file is generated. The EAR file can now be deployed to the IBM Business Monitor server by following standard Java Platform, Enterprise Edition deployment procedures.

Ensure that `WBM_DE.AppTarget` is mapped as the cluster during the deployment as shown in Figure 4-27.

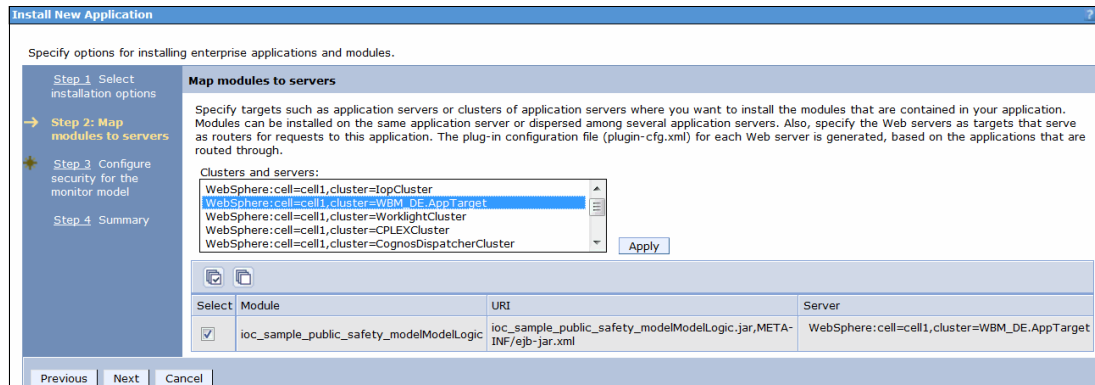


Figure 4-27 Select `WBM_DE.AppTarget` as the cluster during deployment

This completes the Crowd Size Monitor model definition step.

4.5.3 Creating the KPI

In this section, the Crowd Size Monitor model created in 4.5.2, “Defining the monitor model” on page 75 is used to define a new Crowd Size KPI using the IBM Intelligent Operations Center KPI configuration tool.

Create the KPI:

1. Access the Solution Administration view from the toolbar as shown in Figure 4-28.

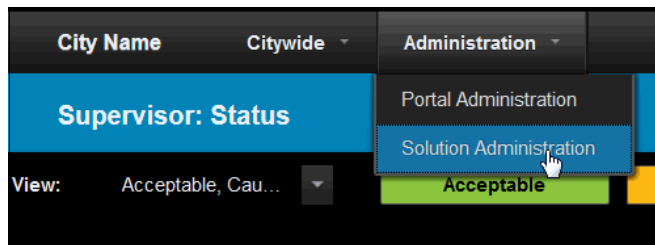


Figure 4-28 Accessing the Solution Administration view

2. Drill down to **Configuration Tool** → **Key Performance Indicators** (Figure 4-29 on page 89).

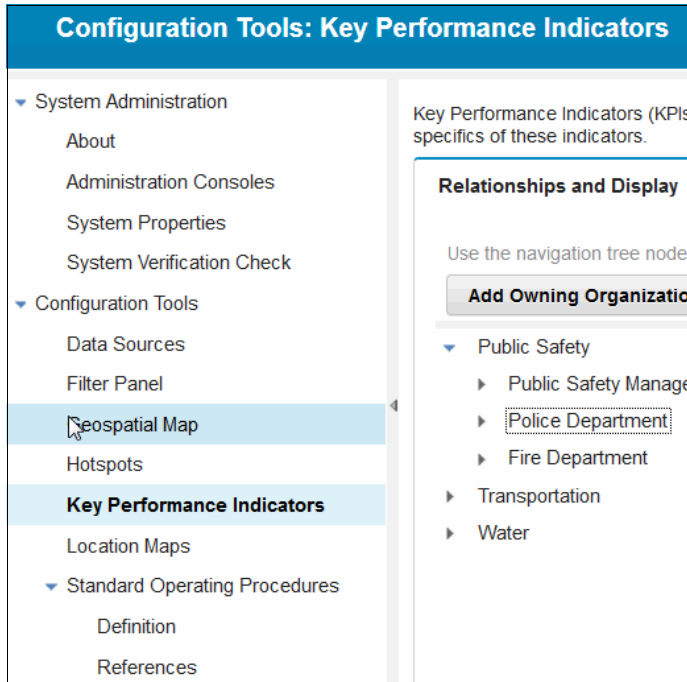


Figure 4-29 Navigating to Key Performance Indicators

3. Click the **KPI Definition** Tab and select **New Aggregate KPI** from the Create pull-down list (Figure 4-30).

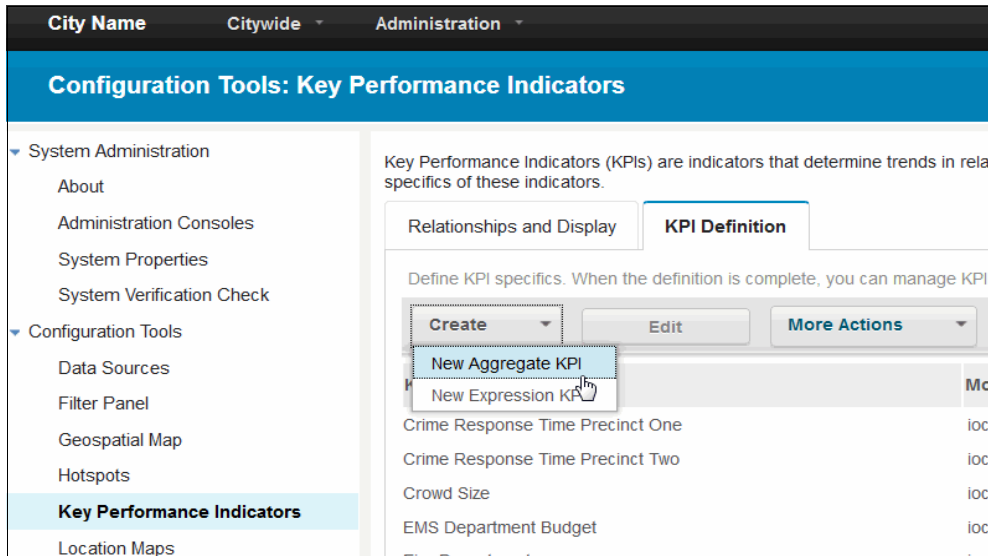


Figure 4-30 Defining the KPI

4. Enter the following information as shown in Figure 4-31 on page 90:
 - For KPI name, type Crowd Size.
 - For Description, type Crowd Size KPI.
 - For Model associated with KPI, select **ioc_sample_public_safety_model**.

Figure 4-31 Setting the KPI properties

5. Click the **Definition** tab and follow the steps (see Figure 4-32):
 - a. Select **Sum** as the operator.
 - b. Select **Crowd Size** as the metric.
 - c. Select **Sliding interval** for Time reference.
 - d. Select **1** for Past, **Hours** for Period, and **Start Date Time** for Time metric.

Figure 4-32 KPI Definition

6. Define the KPI ranges (see Figure 4-33 on page 91):
 - acceptable: 0 -10
 - caution: 10 - 20
 - severe: 20 - 99

Percentage

Range Name	Start Value	End Value	Color	Delete
acceptable	= 0	< 10	Green	X
caution	= 10	< 20	Orange	X
severe	= 20	< 99	Red	X

Buttons: Add row, Copy from legend

Buttons: OK, Cancel

Figure 4-33 Defining the KPI range

7. Click **OK** to create the new KPI.

The new Crowd Size KPI is now created.

4.5.4 Establishing KPI relationships

Next, add the Crowd Size KPI created in 4.5.3, “Creating the KPI” on page 88 to the sample Public Safety owning organization. Follow these steps:

1. Select the **Relationship and Display** tab, right-click **Public Safety**, and select **Add KPI** from the drop-down menu (see Figure 4-34).

Key Performance Indicators (KPIs) are indicators that determine trend the specifics of these indicators.

Relationships and Display | KPI Definition

Use the navigation tree nodes to select KPIs and their definitions.

Buttons: Add Owning Organization, Validate KPI Ranges

- Public Safety
 - Public Safety
 - Police Department
 - Fire Department
 - Transportation
 - Water

Context menu for Public Safety: Add KPI, Edit, Remove

Figure 4-34 Adding a new KPI to the Public Safety organization

2. Select the **Crowd Size** KPI from the drop-down list and click **Add** (see Figure 4-35 on page 92).

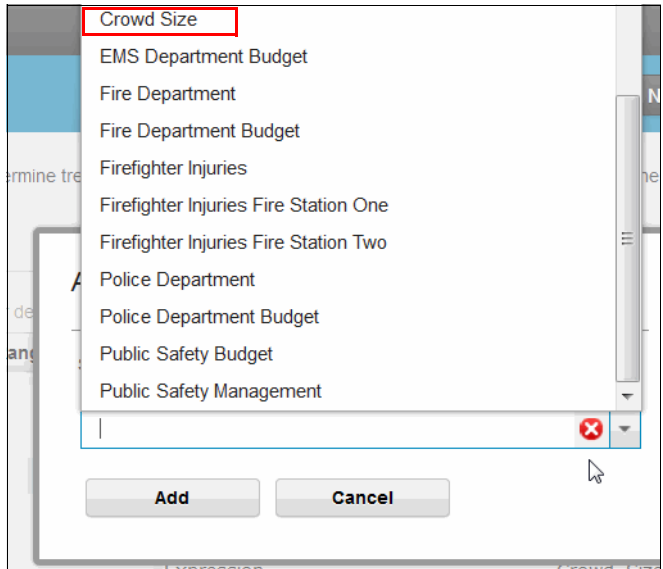


Figure 4-35 Selecting Crowd Size KPI

The Crowd Size KPI is now added as a child of the Police Department KPI as shown in Figure 4-36.

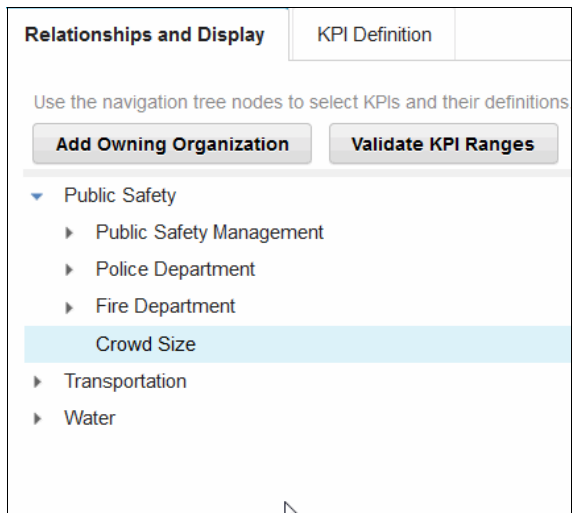


Figure 4-36 Crowd Size KPI added as child of the Public Safety organization

3. Click **Save** (Figure 4-37 on page 93).

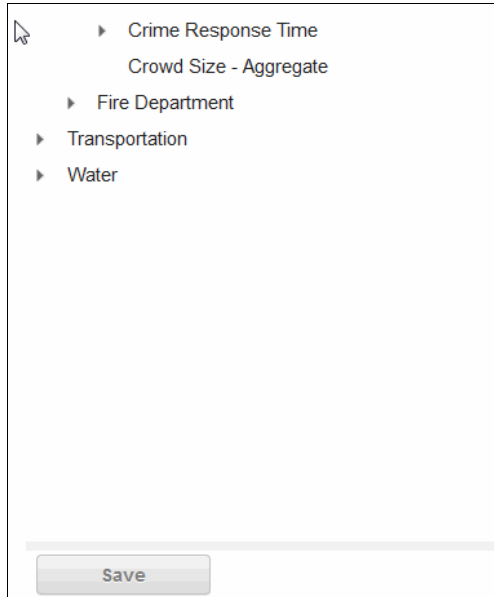


Figure 4-37 Saving the Crowd Size KPI

The Crowd Size KPI is created and added as a child of the Public Safety owning organization.

4.5.5 Visualizing the new KPI

This section describes how to visualize the KPI that added in 4.5.3, “Creating the KPI” on page 88 and 4.5.4, “Establishing KPI relationships” on page 91 using the IBM Intelligent Operations Center dashboard.

Use the status views to obtain a consolidated view of the KPIs. The KPIs that are available in the view depend on the access permission that is assigned to the user. A group of users in an organization is assigned a set of permissions. Members of the group can see only those KPIs that are appropriate to their job. Select the Supervisor:Status or Executive:Status view from the Citywide menu on the top menu bar. Follow these steps:

1. Launch the IBM Intelligent Operations Center dashboard and log in as a user with supervisor privileges.
2. Navigate to the Supervisor:Status view. The Crowd Size KPI is shown with the IBM Intelligent Operations Center sample KPIs under the Public Safety Owning Organization as shown in Figure 4-38 on page 94.

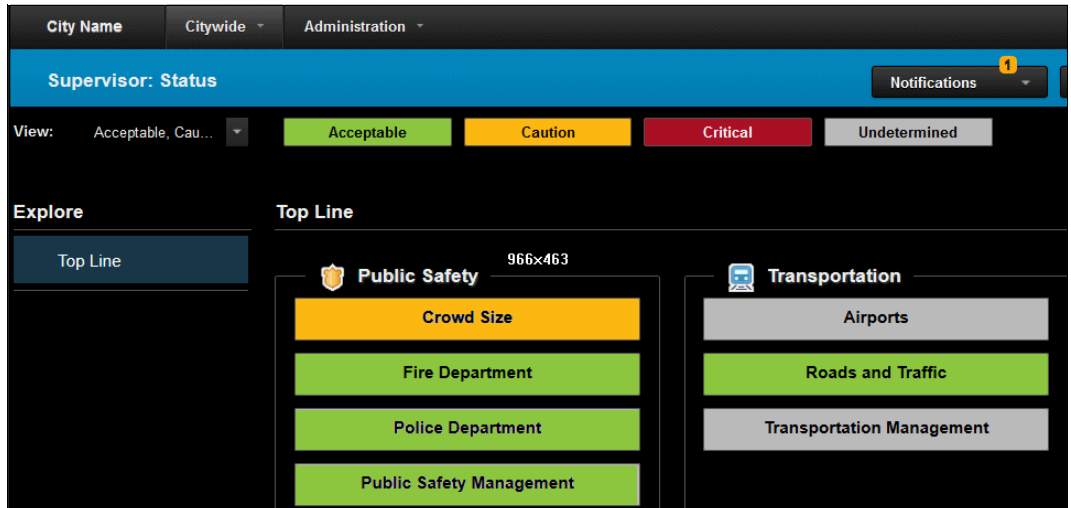


Figure 4-38 Displaying the Crowd Size KPI

4.6 Problem determination

If the KPI configuration tool is not displaying a list of KPIs or models, it usually means that there is a problem communicating with IBM Business Monitor. The problem usually is caused by one of the following reasons:

- ▶ IBM Intelligent Operations Center configuration issues
- ▶ IBM Business Monitor configuration issues
- ▶ IBM Business Monitor credentials issues
- ▶ Secure Sockets Layer (SSL) certificate problem
- ▶ Message routing

4.6.1 IBM Intelligent Operations Center configuration

There are a number of system properties that affect the operation of the KPI infrastructure. System properties are stored in the SYSPROPS table and can be managed using the System Properties configuration tool under the System Administration section as shown in Figure 4-39 on page 95.

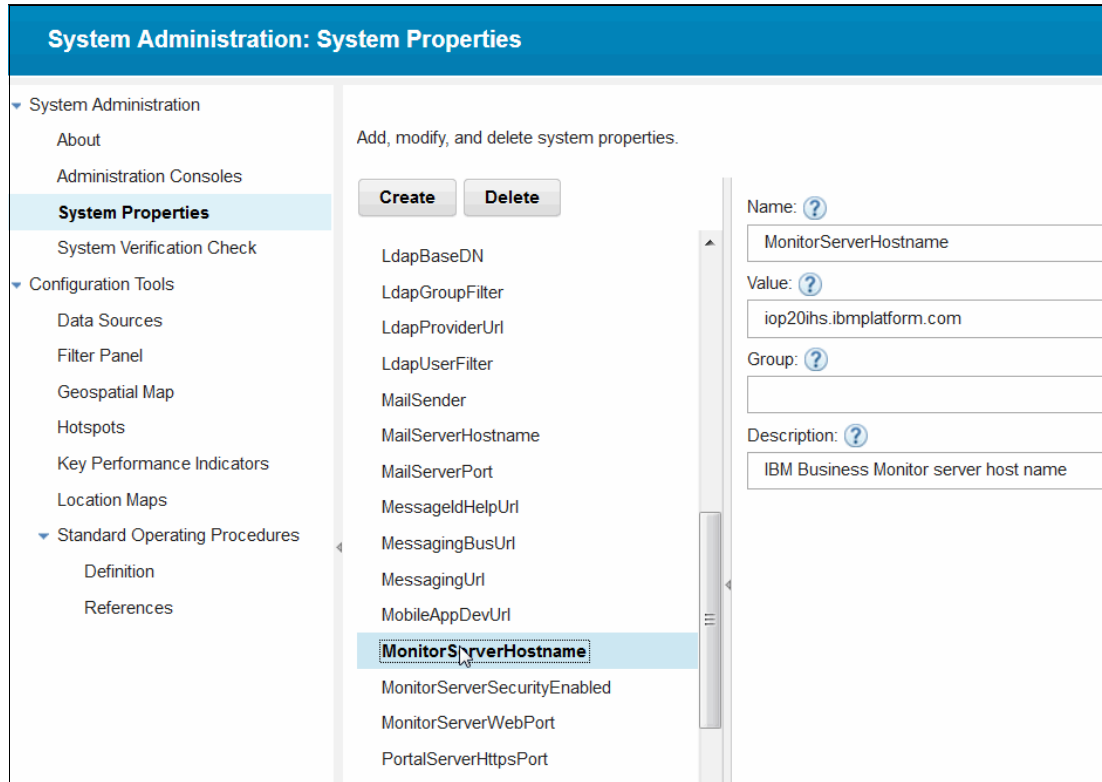


Figure 4-39 Navigating to System Properties

For more information, see *Configuring system properties* at this website:

<http://bit.ly/1mF0qcc>

Table 4-3 on page 96 lists the system properties that affect KPIs.

Table 4-3 KPI-related system properties

Name	Type	Description
MonitorServerHostname	String	Uniform Resource Identifier (URI) for IBM Business Monitor. The value for this property must not include the protocol or the port. See the examples: Correct: wbm.machine.company.com Incorrect: http://www.machine.company.com:9080
MonitorServerWebPort	String	Port to connect to IBM Business Monitor, for example, 9080.
KpiCacheRefreshInterval	Integer	Number of milliseconds between KPI cache refreshes. The default value is 5 minutes.
MonitorServerSecurityEnabled	Boolean	Specifies whether secure connections to the IBM Business Monitor are enabled. If false, http is used. If true, https is used. The protocol must be the same protocol that is supported by the specified MonitorServerWebPort.

4.6.2 IBM Business Monitor configuration

If the IBM Business Monitor models are not deployed correctly or are not running, no models or KPIs will display in the IBM Intelligent Operations Center KPI configuration tool. To correct this error, follow these steps:

1. Log in to the WebSphere Application Server 8.0 Deployment Manager console on the application server with the administrative account. You can access the console from the IBM Intelligent Operations Center portal by clicking **Administration** → **Solution Administration** → **System Administration** → **Administration Consoles** → **Application server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Sametime Proxy).
2. Under Applications, select **Monitor Models**.
3. Ensure that all of the monitor models you require are deployed correctly, have members assigned, and are started as shown in Figure 4-40 on page 97.

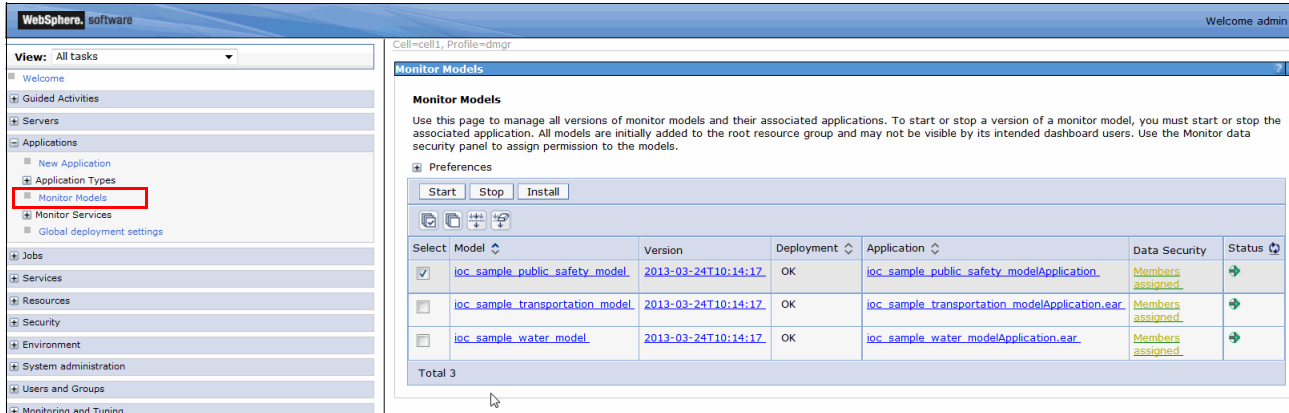


Figure 4-40 Monitor Models started and members assigned

4.6.3 IBM Business Monitor credentials

The KPI portlets use the underlying Java Authentication and Authorization Service (JAAS) to authenticate with the IBM Business Monitor server. The credentials are stored as a JAAS entry with alias `wbadmin` in the WebSphere Application Server administrative console. If you need to modify the credentials, follow these steps:

1. Log in to the WebSphere Application Server administrative console.
2. Navigate to JAAS Security.
3. Find the `wbadmin` alias in the table and click the item.
4. The user ID and password for the `wbadmin` alias can be modified as shown in Figure 4-41.

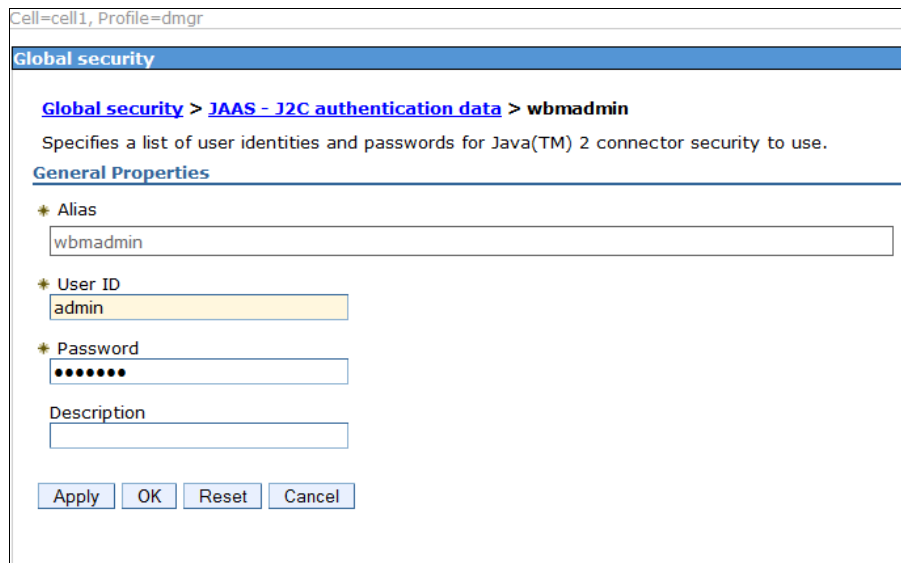


Figure 4-41 JAAS authentication data

5. Restart the portal server for the changes to take effect.

4.6.4 IBM Business Monitor SSL certificates

To create secure connections from IBM Intelligent Operations Center to IBM Business Monitor, IBM Intelligent Operations Center needs to maintain an SSL certificate corresponding to the IBM Business Monitor instance used.

If there are SSL certificate errors in the IBM Intelligent Operations Center logs, a possible reason is that the SSL certificate is incorrect. Follow these steps to retrieve the SSL certificate:

1. Log in to the WebSphere Application Server administrative console.
2. Navigate to the **Security** section and select **SSL certificate and key management** (Figure 4-42).



Figure 4-42 SSL certificate and key management

3. Select **Key stores and certificates**.
4. Click the **NodeDefaultTrustStore** link corresponding to the portal node (Figure 4-43 on page 99).

SSL certificate and key management > Key stores and certificates
 Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all truststore types.

Keystore usages
 SSL keystores

Preferences
 New... Delete Change password... Exchange signers...

Select	Name	Description	Management Scope	Path
<input type="checkbox"/>	CMSKeyStore	CMSKeyStore for web server ihserver1.	(cell):cell1:(node):ihserver1:(server):ihserver1	\${CONFIG_ROOT}/cells/cell1/nodes/ihserver1/servers/ihserver1/plugin-key.kdb
<input type="checkbox"/>	CellDefaultKeyStore	Default key store for cell1	(cell):cell1	\${CONFIG_ROOT}/cells/cell1/key.p12
<input type="checkbox"/>	CellDefaultTrustStore	Default trust store for cell1	(cell):cell1	\${CONFIG_ROOT}/cells/cell1/trust.p12
<input type="checkbox"/>	NodeDefaultKeyStore	Default key store for CognosNode1	(cell):cell1:(node):CognosNode1	\${CONFIG_ROOT}/cells/cell1/nodes/CognosNode1/key.p12
<input type="checkbox"/>	NodeDefaultKeyStore	Default key store for PortalNode1	(cell):cell1:(node):PortalNode1	\${CONFIG_ROOT}/cells/cell1/nodes/PortalNode1/key.p12
<input type="checkbox"/>	NodeDefaultKeyStore	Default key store for IopNode1	(cell):cell1:(node):IopNode1	\${CONFIG_ROOT}/cells/cell1/nodes/IopNode1/key.p12
<input type="checkbox"/>	NodeDefaultKeyStore	Default key store for CPLEXNode1	(cell):cell1:(node):CPLEXNode1	\${CONFIG_ROOT}/cells/cell1/nodes/CPLEXNode1/key.p12
<input type="checkbox"/>	NodeDefaultKeyStore	Default key store for WBMNode1	(cell):cell1:(node):WBMNode1	\${CONFIG_ROOT}/cells/cell1/nodes/WBMNode1/key.p12
<input type="checkbox"/>	NodeDefaultKeyStore	Default key store for WorklightNode1	(cell):cell1:(node):WorklightNode1	\${CONFIG_ROOT}/cells/cell1/nodes/WorklightNode1/key.p12
<input type="checkbox"/>	NodeDefaultKeyStore	Default key store for STProxyNode1	(cell):cell1:(node):STProxyNode1	\${CONFIG_ROOT}/cells/cell1/nodes/STProxyNode1/key.p12
<input type="checkbox"/>	NodeDefaultTrustStore	Default trust store for CognosNode1	(cell):cell1:(node):CognosNode1	\${CONFIG_ROOT}/cells/cell1/nodes/CognosNode1/trust.p12
<input type="checkbox"/>	NodeDefaultTrustStore	Default trust store for PortalNode1	(cell):cell1:(node):PortalNode1	\${CONFIG_ROOT}/cells/cell1/nodes/PortalNode1/trust.p12

Figure 4-43 Selecting NodeDefaultTrustStore for portal node

5. Select **Signer certificates** (Figure 4-44).

SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore
 Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all truststore types.

General Properties

Name
 NodeDefaultTrustStore

Description
 Default trust store for PortalNode1

Management scope
 (cell):cell1:(node):PortalNode1

Path
 \${CONFIG_ROOT}/cells/cell1/nodes/PortalNode1/trust.p12

Additional Properties

- [Signer certificates](#)
- [Personal certificates](#)
- [Personal certificates](#) Manage
- [Custom properties](#)

Figure 4-44 Selecting Signer certificates

6. Click **Retrieve from port** (Figure 4-45 on page 100).

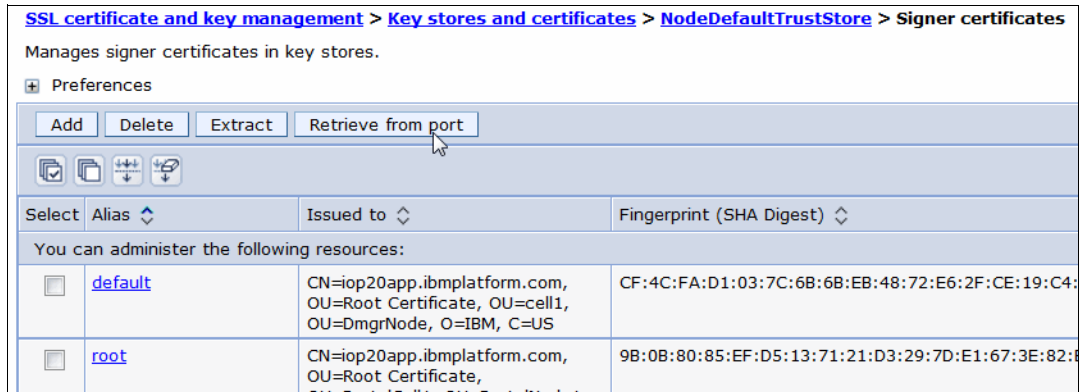


Figure 4-45 Retrieve from port

7. Enter the URL of the new IBM Business Monitor instance and a port number that is open for SSL connections. You must also provide a unique alias for this connection (Figure 4-46).

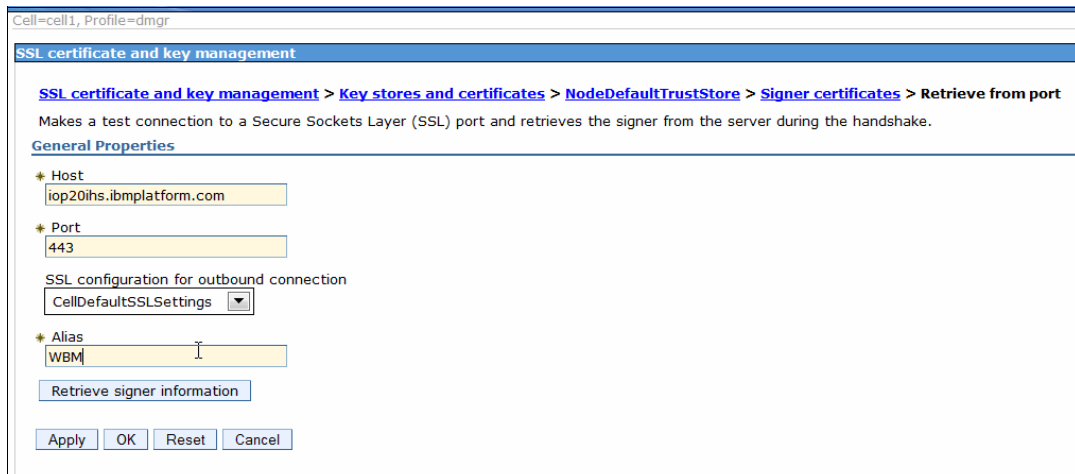


Figure 4-46 Retrieve signer information

If the correct details are supplied, the signer information is populated as shown in Figure 4-47 on page 101.

[SSL certificate and key management](#) > [Key stores and certificates](#) > [NodeDefaultTrustStore](#) > [Signer](#)

Makes a test connection to a Secure Sockets Layer (SSL) port and retrieves the signer from the server

General Properties

* Host
iop20ihs.ibmplatform.com

* Port
443

SSL configuration for outbound connection
CellDefaultSSLSettings

* Alias
WBM

Retrieve signer information

Retrieved signer information

Serial number
1375439030

Issued to
CN=iop20ihs.ibmplatform.com, O=IBM, OU=IOC, C=US

Issued by
CN=iop20ihs.ibmplatform.com, O=IBM, OU=IOC, C=US

Fingerprint (SHA digest)
EC:92:A0:E7:AB:01:3B:22:49:EE:9B:37:3B:EC:13:30:00:04:A7:26

Validity period
31/07/2023

Apply OK Reset Cancel

Figure 4-47 Successfully retrieved signer information

8. Click **OK** to save the changes.

4.6.5 Message routing

To determine whether a message has been routed from IBM Intelligent Operations Center to IBM Business Monitor, turn tracing on for the data receiver component:

1. Log in to the WebSphere Application Server administrative console.
2. Click **Troubleshooting Logs and Trace** → **WebSphere_Portal** → **Change log level details**.
3. Click the Runtime tab and enter the following command:

```
*=warning: com.ibm.ioc.logger.DataReceiverLogger=all
```

To determine whether the messages were received by IBM Business Monitor, turn on the IBM Business Monitor event recording. To turn on event recording, follow these steps:

1. Log in to the WebSphere Application Server administrative console.
2. Navigate to **Applications** → **Monitor Services** → **Recorded Event Management** → **Event Recording**.
3. To view the recorded events, select **Applications** → **Monitor Services** → **Recorded Event Management** → **Event Management**.

4.7 Programmatic access to KPIs

IBM Intelligent Operations Center uses IBM Business Monitor as its KPI processing engine. KPIs are defined in IBM Business Monitor model applications, which are installed on the IBM Business Monitor server (IBM Intelligent Operations Center application server).

IBM Intelligent Operations Center retrieves the KPI values from the IBM Business Monitor server at startup and stores the values in cache. The cache is refreshed at configurable intervals to ensure that the KPI values are current. KPIs can also be updated asynchronously by adding outbound events to the monitor model applications that send KPI updates to the IBM Intelligent Operations Center based on various trigger conditions.

Regardless of how the KPIs are updated, the KPI values are stored in the cache and made available to consuming applications via the KPI service. IBM Business Monitor allows you to define KPIs whose values are derived from other KPIs but it does not support hierarchical KPIs. IBM Intelligent Operations Center implements hierarchical KPIs by overlaying a hierarchical structure onto a set of KPIs where the parent KPI values are defined to derive their value from one or more child KPIs.

The KPI service is used to define the KPI hierarchy. The hierarchy consists of owning organizations and KPIs. An owning organization is a container object that is associated with a monitor model, an icon, and a display label. The owning organization contains one or more child KPIs defined in its associated monitor model. The child KPIs can themselves have child KPIs. For more information about the KPI service, see the *IBM Intelligent Operations Center V1.6 REST APIs* document.

IBM Intelligent Operations Center provides a set of REST APIs to programmatically interface with the KPI service. This section includes typical request and response examples using the KPI REST service:

- ▶ Example 4-2 shows the request and response to retrieve cached KPI values.

Example 4-2 Retrieve cached KPI values

```
Request URL: https://<ioc_ihs>/ibm/ioc/api/kpi-service/kpis
Operation Type: GET
Tool: Chrome REST Client
JSON Response:
{
  "results": [{
    "parentId": "2a66cf8f-9a3f-46d2-8267-7d1d58206cf2",
    "rangeColor": "#fdb813",
    "rangeType": "actualValue",
    "rangeId": "caution",
    "rangeName": "Caution",
    "hasChildren": false,
```



```

        "dbId": 4,
        "kpiCalcMethod": "aggregated",
        "name": "Water Leakage",
        "id": "0518e2e5-1195-4aff-867f-5294e4072344",
        "entryType": "kpi",
        "kpiValue": 150.0
    },
    {
        "parentId": "f6d9aa9b-27f7-46b3-8f5b-8773ccb7ee41",
        "rangeColor": "#8cc63f",
        "rangeType": "actualValue",
        "rangeId": "acceptable",
        "rangeName": "Acceptable",
        "hasChildren": true,
        "dbId": 9,
        "kpiCalcMethod": "calculated",
        "name": "Flood Control",
        "id": "21e133b9-b678-4323-8d8a-e6882b6b04e2",
        "entryType": "kpi",
        "kpiValue": 0.0
    }
  ]],
  "lastupdate": 1379253069206
}

```

- Manage the KPI hierarchy. Example 4-3 shows the request and response to retrieve the KPI hierarchy.

Example 4-3 Retrieve KPI hierarchy

This sample retrieves the child level KPIs for owning organization with identifier of f6d9aa9b-27f7-46b3-8f5b-8773ccb7ee41

Request URL:

https://<ioc_ihs>/ibm/ioc/api/kpi-service/kpis?owningorg=f6d9aa9b-27f7-46b3-8f5b-8773ccb7ee41

Operation Type: GET

Tool: Chrome REST Client

JSON Response:

```

{
  "results": [{
    "owningOrgId": "f6d9aa9b-27f7-46b3-8f5b-8773ccb7ee41",
    "parentId": "f6d9aa9b-27f7-46b3-8f5b-8773ccb7ee41",
    "rangeColor": "#8cc63f",
    "rangeName": "Acceptable",
    "name": "Flood Control",
    "entryType": "kpi",
    "kpiValue": 0.0,
    "rangeType": "actualValue",
    "rangeId": "acceptable",
    "dbId": 27,
    "hasChildren": true,
    "kpiCalcMethod": "calculated",
    "id": "21e133b9-b678-4323-8d8a-e6882b6b04e2"
  }],
  {
    "owningOrgId": "f6d9aa9b-27f7-46b3-8f5b-8773ccb7ee41",
    "parentId": "f6d9aa9b-27f7-46b3-8f5b-8773ccb7ee41",

```

```

        "rangeColor": "#fdb813",
        "rangeName": "Caution",
        "name": "Water Management",
        "entryType": "kpi",
        "kpiValue": 1.0,
        "rangeType": "actualValue",
        "rangeId": "caution",
        "dbId": 36,
        "hasChildren": true,
        "kpiCalcMethod": "calculated",
        "id": "456ed1bd-d51c-4b79-8dbf-bea2526cfa32"
    }
}

```

- Manage the owning organization. Example 4-4 shows the request and response to retrieve the owning organization.

Example 4-4 Sample owning organization call

```

Request URL: https://<ioc_ihs>/ibm/ioc/api/kpi-service/owningorgs
Operation Type: GET
Tool: Chrome REST Client
JSON Response:
[
  {
    "hasChildren": true,
    "icon": "/ibm/ioc/theme/static/js/images/owning_org_icons/safety_24.png",
    "dbId": 0,
    "name": "Public Safety",
    "id": "424109cd-8455-4c12-8fdc-111d8aa92316",
    "entryType": "model"
  },
  {
    "hasChildren": true,
    "icon":
"/ibm/ioc/theme/static/js/images/owning_org_icons/transportation_24.png",
    "dbId": 1,
    "name": "Transportation",
    "id": "3b05b9fc-ecbc-445b-88ee-823d6ed073a1",
    "entryType": "model"
  }
]

```

The KPI service also provides a proxy to the underlying IBM Business Monitor KPI definitions. The KPI definition resource can forward KPI management requests (query, create, update, and delete) to the IBM Business Monitor KPI definition REST interface. See Example 4-5 to see the proxy service in action.

Example 4-5 KPI definition proxy call

To obtain a listing of all models deployed, you can invoke the following REST call directly against the Monitor Server

```

Request URL: https://<ioc_app>/rest/bpm/monitor/models/
Operation Type: GET

```

The proxy KPI definition can be used to invoke the same call:

Request URL:
https://<ioc_>ibm/ioc/api/kpi-service/kpi-definition/rest/bpm/monitor/models/
Operation Type: GET

The response from both calls will be identical like this

Tool: Chrome REST Client

JSON Response:

```
[{
  "Model ID": "icoc_sample_public_safety_monitor_model",
  "Versions": [{
    "Model Display Name": "icoc sample public safety monitor model",
    "Version": 20110218104946
  }]
},
{
  "Model ID": "icoc_sample_transportation_monitor_model",
  "Versions": [{
    "Model Display Name": "icoc sample transportation monitor model",
    "Version": 20110218104946
  }]
},
{
  "Model ID": "icoc_sample_water_monitor_model",
  "Versions": [{
    "Model Display Name": "icoc sample water monitor model",
    "Version": 20110218104946
  }]
},
{
  "Model ID": "ioc_15_weather_monitor_model",
  "Versions": [{
    "Model Display Name": "ioc_15_weather_monitor_model",
    "Version": 20120928115248
  }]
}]
```

For more information about the IBM Business Monitor KPI definition REST interface, see *IBM Business Monitor REST interface*:

<http://bit.ly/1fudt8F>

4.8 Additional considerations

This section includes miscellaneous information to help you implement KPIs successfully.

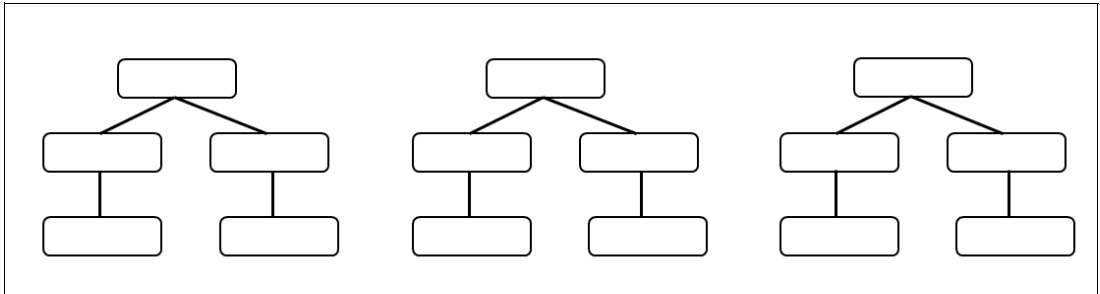


Figure 4-49 Good KPI hierarchy



Driving standard operating procedures

Managing the response to city events and incidents is always a complex task. Standard operating procedures (SOPs) are a structured way to plan, monitor, and execute coordinated efforts to respond to events and incidents. SOP applications are not limited to cities. They apply to any type of organization with these requirements. SOP activities can be tracked and integrated in the IBM Intelligent Operations Center user interface.

This chapter provides an overview of the SOP support in IBM Intelligent Operations Center. It includes use case scenarios to illustrate practical implementation examples.

This chapter includes the following topics:

- ▶ 5.1, “Managing SOPs in IBM Intelligent Operations Center” on page 110
- ▶ 5.2, “Configuring activities for different scenarios” on page 125
- ▶ 5.3, “SOP REST service” on page 144
- ▶ 5.4, “Use case overview” on page 154
- ▶ 5.5, “Manage event SOP” on page 156
- ▶ 5.6, “Manage crowd incident SOP” on page 177

5.1 Managing SOPs in IBM Intelligent Operations Center

An *SOP* is a set of instructions that describe all the relevant steps and activities of a process or procedure. In the context of cities, SOPs are a powerful tool to organize the process of responding to events and incidents in a coordinated manner. This approach enables the city managers and involved agencies to organize personnel, information, and tasks to achieve comprehensive control of the operation.

IBM Intelligent Operations Center provides a collaborative environment in which different agencies can manage SOPs to respond to events and incidents in the city in a seamless manner. IBM Intelligent Operations Center also provides ways to extend and customize the SOP functions to address specific client requirements.

SOPs are defined by city officials and IBM Intelligent Operations Center provides full capabilities for a business user with no technical background to create and execute SOPs. If the client requirements call for the extension of the standard SOP capabilities provided with IBM Intelligent Operations Center, the available service interfaces can be used to programmatically extend the SOP capabilities.

An SOP definition during its lifecycle can assume different statuses:

- ▶ **Draft:** When the SOP is first created, a draft version is saved initially. From an approved version of an SOP, it is also possible to create another draft version, when it is necessary to change the SOP definition using the approved version as a base. A draft can be edited, submitted for approval, or discarded.
- ▶ **Pending approval:** This is a draft SOP definition submitted for approval that is ready to be approved or discarded. The name of the version is defined in this status and it will name the SOP definition version if approved. If this version is not approved, the SOP definition goes back to the draft version status.
- ▶ **Approved:** When an SOP definition is approved, it is ready to be launched.

Figure 5-1 shows the SOP approval flow.

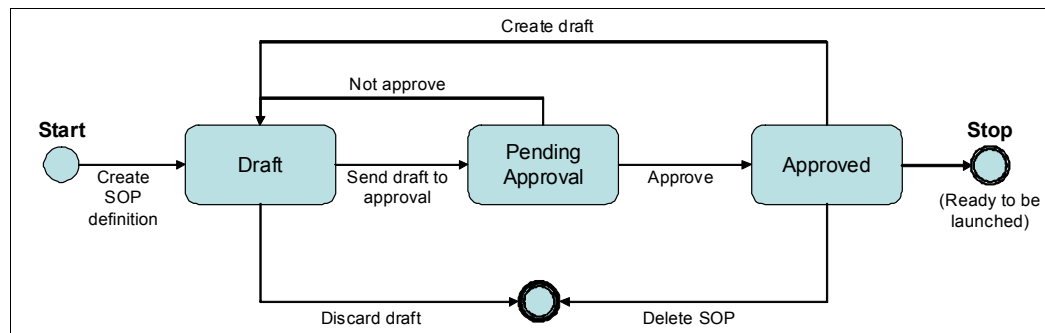


Figure 5-1 SOP approval lifecycle

The SOP approval flow occurs before the launching of the SOP. The process is important because creating an SOP is a task that involves different agencies and different levels of organization that, after a number of interactions, must agree on a final version of the procedure.

An activity within an SOP or the entire SOP can run for a long time. Changes can be applied to an SOP definition while one of its instances is still in execution. For example, a new version of an SOP (for example, v2) can be approved while the previous version is still running (v1).


This approval flow makes it possible to create a new SOP definition without affecting the running instances.

5.1.1 Roles and permissions

This section explains the configurations required to grant users the permissions they need for various SOP-related actions.

Granting access to the SOP Definition portal page

Users that need to create and edit SOP definitions and activities must have access to the SOP configuration tool portal pages in the Solution Administrator user interface (UI). Grant user groups access to the portal pages:

1. Log in to the IBM Intelligent Operations Center portal as `wpsadmin`.
2. Click **Administration** → **Portal Administration**.
3. Click **Manage Pages** and navigate to **Content Root** → **Solution Administration** → **Configuration Tools** → **Standard Operating Procedures**.
4. Click the **Set Page Permission** icon  for the Definition page (see Figure 5-2).

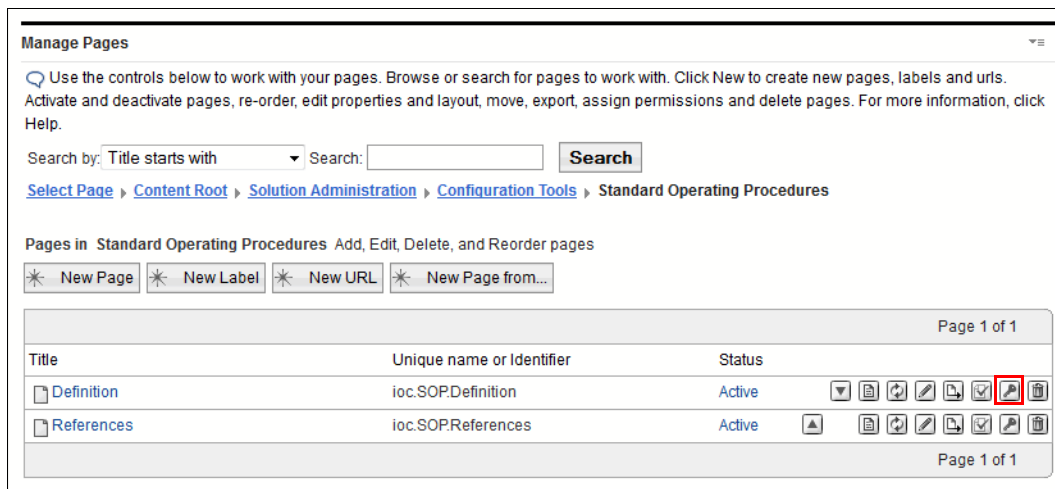


Figure 5-2 Granting access to SOP Definition portal page

5. In the Resource Permissions window, click the **Edit role** icon for the User role and then click **Add** (see Figure 5-3).

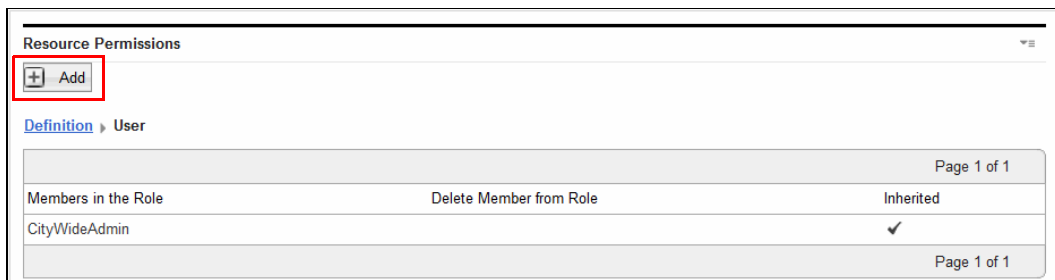


Figure 5-3 Adding portal users to the User role for the SOP Definition portal page

Note: By default, users in the CityWideAdmin group have user access to the SOP Definition page. You can simply add users to this group to grant them access to this page.

6. Click **Search** and locate the user or group to which you want to grant access to the SOP Definition page.
7. Select the user or group and click **OK**.

Figure 5-4 shows as an example, granting the CityWideSupervisor group access to the SOP Definition portal page.

The screenshot shows a 'Resource Permissions' dialog box. At the top, there is a search bar with the text 'Search for Users or User Groups.' and a dropdown menu set to 'User Groups'. Below the search bar, there is a 'Search by:' dropdown set to 'All available' and a 'Search' button. The main area of the dialog is a list of user groups with checkboxes. The 'CityWideSupervisor' group is selected, indicated by a checked checkbox and a blue highlight. Other groups listed include 'Users and User Groups', 'All Authenticated Portal Users', 'All Portal User Groups', 'wpsDocReviewer', 'wpsadmins', 'CityWideOperator', 'wpsContentAdministrators', 'CityWideAdmin', 'stadmins', and 'CityWideExecutive'. At the bottom of the dialog, there are 'OK' and 'Cancel' buttons. The dialog also features pagination controls at the top right and bottom right, both showing 'Page 1 of 2' and 'Jump to page: 1'.

Figure 5-4 Granting CityWideSupervisor group user access to the SOP Definition portal page

Adding user groups to system properties

To grant users some of the permissions required to configure and manage SOPs, you need to add user groups to one or more of the following system properties:

- ▶ SopAdminRoles
- ▶ SopAuthorRoles
- ▶ ReferenceLibrarianRoles

Note: You cannot add user IDs to the system properties; you must add user groups.

For example, add the group CityWideAdmin to the SopAuthorRoles system property:

1. Log in to the IBM Intelligent Operations Center portal as wpsadmin.
2. Click **Administration** → **Solution Administration**.
3. Expand **System Administration** and click **System Properties**.
4. Scroll down to find the **SopAuthorRoles** system property and click it.
5. Add the **CityWideAdmin** group as shown in Figure 5-5 and click **Save**.

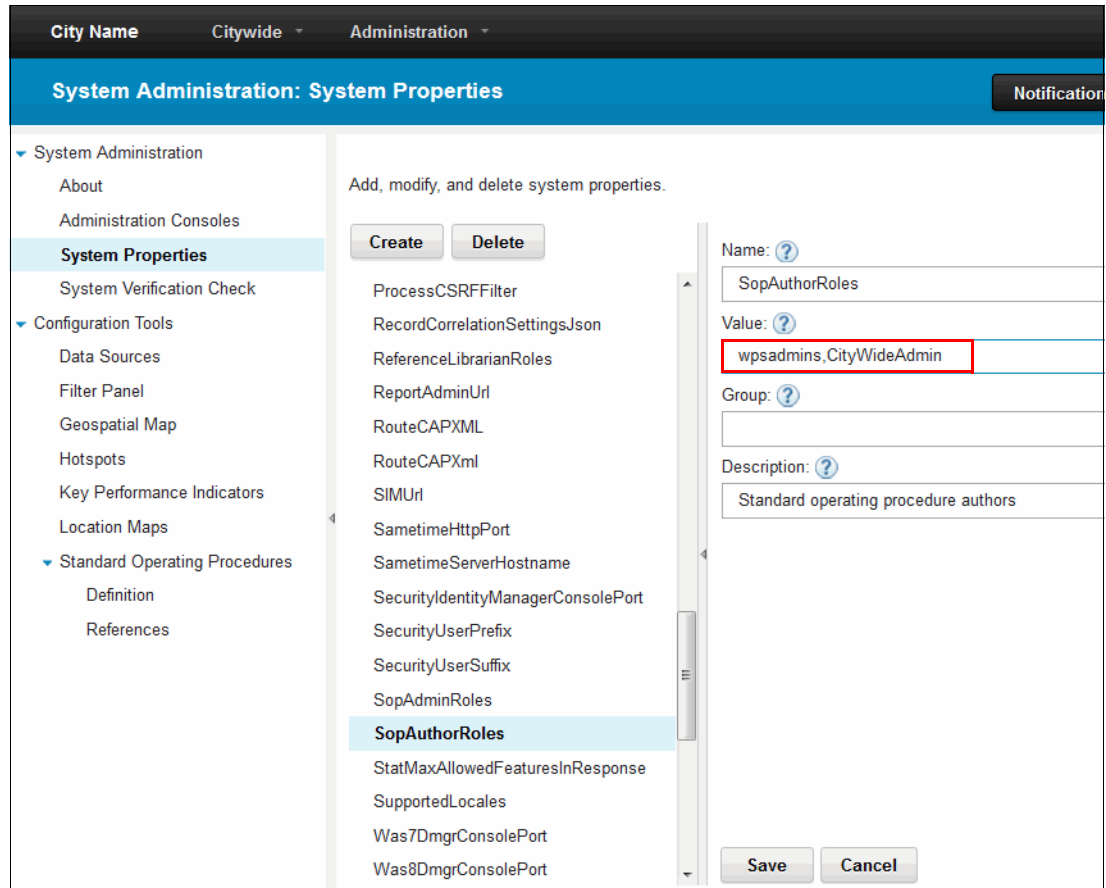


Figure 5-5 Adding a group (CityWideAdmin) to the SopAuthorRoles system property

Note: Changes to the SopAuthorRoles, SopAdminRoles, and ReferenceLibrarianRoles take place immediately. You do not need to restart the portal cluster.

Restarting the portal cluster

You do not need to restart the portal cluster for changes to the SopAuthorRoles, SopAdminRoles, and ReferenceLibrarianRoles to take effect.

This section includes the steps to restart the portal cluster if necessary:

1. Log in to the IBM Intelligent Operations Center portal as wpsadmin.
2. Access the WebSphere Application Server Integrated Solution Console:
 - a. Click **Administration** → **Solution Administration** → **System Administration** → **Administration Consoles**.
 - b. Click **Application Server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Sametime Proxy).

- c. Log in as administrator.
3. Navigate to **Servers** → **Clusters** → **WebSphere application server clusters**. Select **PortalCluster** and click **Stop**.

Figure 5-6 shows stopping the portal cluster.

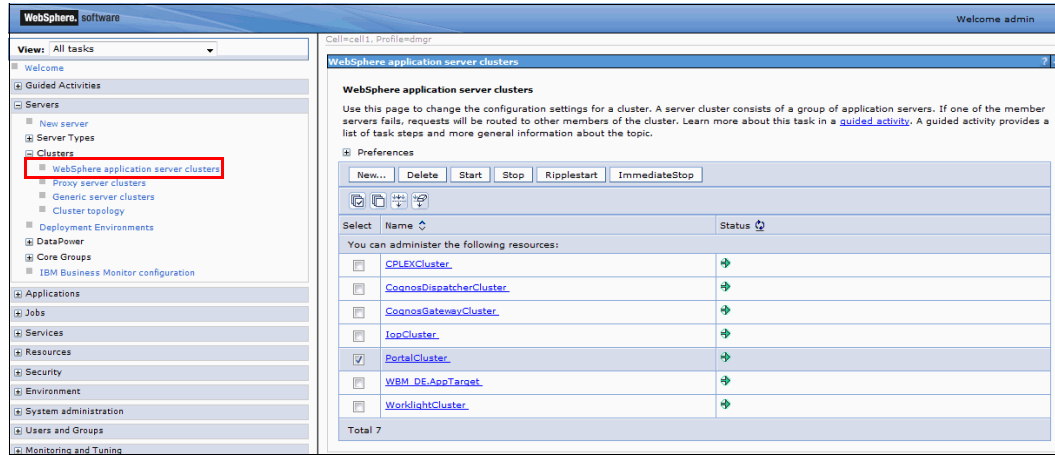


Figure 5-6 Stopping the PortalCluster

4. Refresh the status and wait until PortalCluster is stopped.
5. Click **Start** to start PortalCluster.

Defining SOP roles

Table 5-1 on page 114 outlines the roles required for SOP-related actions.

Note: A user needs to be in at least one of the roles indicated as “Yes” in Table 5-1 to be able to perform the action.

Table 5-1 SOP actions and required roles

Action	SopAdminRoles	SopAuthorRoles	ReferenceLibrarianRoles	Owners (SOP definition)	Readers (SOP definition)	Owners (SOP activity)	Readers (SOP activity)
Create SOP definition	No	Yes	No	No	No	No	No
Create SOP draft	No	Yes	No	Yes	No	No	No
View SOP definition	Yes	Yes	No	Yes	Yes	No	No
Edit SOP definition	No	Yes	No	Yes	No	No	No
Delete SOP definition	Yes	Yes	No	Yes	No	No	No

Action	SopAdminRoles	SopAuthorRoles	ReferenceLibrarianRoles	Owners (SOP definition)	Readers (SOP definition)	Owners (SOP activity)	Readers (SOP activity)
View SOP activity	No	Yes	No	No	Yes, provided that the user has the Readers role in the activity definition	No	No
Edit SOP activity	No	Yes	No	Yes	No	No	No
Delete SOP activity	No	Yes	No	Yes	No	No	No
Submit SOP draft for approval	No	Yes	No	Yes	No	No	No
Approve SOP draft	No	Yes	No	Yes	No	No	No
Launch SOP instance	Yes	No	No	Yes	No	No	No
View SOP instance	Yes	No	No	Yes	Yes from My Activities	Yes from My Activities	Yes from My Activities
Edit SOP instance	Yes	No	No	Yes	No	No	No
Start and complete activities in SOP instance	Yes	No	No	No	No	Yes from My Activities For own activities	No
Create shared references	No	No	Yes	No	No	No	No

For more information, see *Defining standard operating procedure roles* at this website:

<http://bit.ly/1nJWIve>

5.1.2 Configuring SOP definitions

When you define an SOP, you define activities that are included in the SOP. You can specify the order in which some or all of the activities are executed. For example, you can specify that a particular activity is not started until the previous activity is completed or skipped.

To create a new SOP definition, follow these steps:

1. In the Solution Administration view, click **Configuration Tools** → **Standard Operating Procedures** → **Definition**.

2. Click **Create**.
3. Follow the steps in the Create Standard Operating Procedure Definition wizard to create a first draft.

You can assign roles to user groups on the Roles tab of the SOP definition as shown in Figure 5-7. The columns Owners (SOP definition) and Readers (SOP definition) in Table 5-1 on page 114 show the actions that these user groups are authorized to perform.

Create Standard Operating Procedure Definition

Basics **Roles** Activities Summary

Select the roles that are responsible for monitoring the execution and completion of this procedure.

Available Roles:

Enter a filter string.

- CityWideExecutive
- CityWideSupervisor
- stadmins
- wcmadmins
- wpsadmins
- wpsContentAdministrators
- wpsDocReviewer

Selected Roles:

Role	Owners	Readers
CityWideAdmin	<input checked="" type="radio"/>	<input type="radio"/>
CityWideOperator	<input checked="" type="radio"/>	<input type="radio"/>

Add **Remove**

Back **Next** **Cancel**

Figure 5-7 SOP definition roles

The SOP definition draft, after it is saved, is ready to be submitted for approval. When it is in the draft version, the SOP definition can be edited at any time, before submitting it for approval.

For more information, see *Configuring standard operating procedure definitions*:

<http://bit.ly/1fbyDhr>

5.1.3 Editing definitions

To change an SOP definition, it is required that you create a draft first (an approved version cannot be edited directly).

To make changes to an approved SOP version, follow these steps:

1. In the SOP definitions list, click the name of the SOP to be changed.
2. From the Version pull-down list, select a version.

3. Click **Create Draft** and then select **OK**.
4. Follow these steps to make changes:
 - a. Select **Edit** to edit basic information.
 - b. To edit activities, select the **Activities** tab, select the activity and click **Edit** or **Delete** to update or remove the activity or click **Add** to add a new activity to the activities list.
 - c. To edit roles, select the **Roles** tab and then click **Edit**.
 - d. To edit references, select the **References** tab, click **Edit** or **Delete** for a specific reference or click **Add** to add a new reference.

Figure 5-8 shows editing the SOP activities to make changes.

Figure 5-8 Editing an SOP

5. After you complete editing the SOP draft, click **Save** to save the draft.

5.1.4 Working with drafts and submitting for approval

When working with drafts that have not been submitted for approval, you can submit the draft for approval or discard it:

- ▶ To submit a draft for approval, select **Submit for Approval** in the Submit for Approval list. The SOP is assigned a version name, which is the time stamp when it was submitted for approval. If the SOP is approved, this name will be the final name of the version.
- ▶ To discard a draft, select **Discard Draft** in the Submit for Approval list.

Figure 5-9 shows the Submit for Approval and Discard Draft options in the Submit for Approval list.

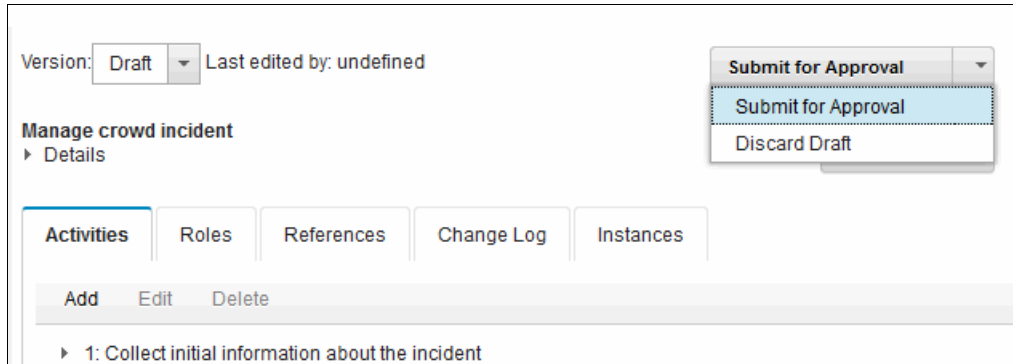


Figure 5-9 Submitting for approval or discarding an SOP draft

5.1.5 Approving or disapproving drafts

Draft SOPs submitted for approval can be approved or disapproved. To approve or disapprove a new SOP or changes to an SOP definition, users must have the roles that are required to approve an SOP draft as described in Table 5-1 on page 114.

Note: SOPs must go through a formal approval process before they are approved in IBM Intelligent Operations Center. There needs to be a well-defined process for selecting required reviewers, sending comments to the originator, and approving the SOP draft. The staff responsible for approving the SOP needs to include a representative of the group of individuals who will be affected by the SOP.

To approve or disapprove a new SOP or changes to an SOP, follow these steps:

1. In the SOP definitions list, click the name of the SOP that has been submitted for approval.
2. To approve a version, select **Approve** from the Approve list. At this point, the SOP is ready to be launched.
3. To disapprove a draft, select **Disapprove** from the Approve list. The SOP definition changes back to the Draft status.

Figure 5-10 on page 118 shows the Approve and Disapprove options in the Approve list.

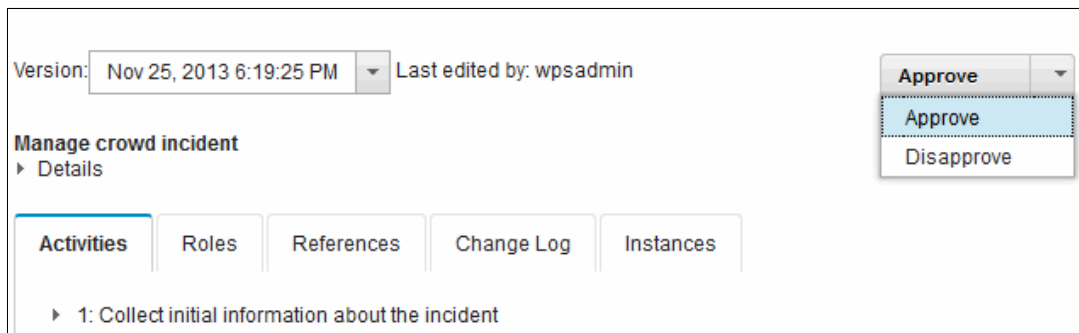


Figure 5-10 Approving or disapproving an SOP draft

5.1.6 Launching SOPs

When an SOP definition is approved, it is ready to be launched. SOPs can be launched in one of the following ways:

- ▶ Automatically from a routing expression in a data source
- ▶ Manually by the operator from the SOP configuration tool in the Solution Administration view
- ▶ Manually by the operator from the More actions pull-down menu in an event preview card
- ▶ Via a programmatic interface as explained in 5.3.4, “Working with SOP instances” on page 148

Launching an SOP from a routing expression in a data source

You can configure a routing expression in a data source to launch an SOP. IBM Intelligent Operations Center launches the SOP when a data item that meets the criteria specified in the routing expression is ingested.

For information about creating data sources, see Chapter 3, “Integrating with data sources” on page 45.

Figure 5-11 shows a routing expression to launch the Manage crowd incident SOP when the CRWDSIZE property in the data item is greater than 20.

▼ Routing Expression		
* Property: ?	* Operation: ?	* Criteria: ?
CRWDSIZE	Greater than	20
* Route to: ?	* Standard operating procedure: ?	
Standard Operating Procedures	Manage crowd incident	

Figure 5-11 Routing expression to launch the Manage crowd incident SOP

Launching SOPs from the Solution Administration

To launch an SOP from the SOP configuration tool in the Solution Administration, follow these steps:

- ▶ The user must have one of the roles required to launch an SOP instance as described in Table 5-1 on page 114. The user must have one of the following roles:
 - User group is in the SopAdminRoles system property.
 - User group has access type Owner in the SOP definition.
- ▶ The user is authorized to the Standard operating procedures → Definition portal page as described in “Granting access to the SOP Definition portal page” on page 111.

Launch the SOP from the Solution Administration:

1. In the Solution Administration view, click **Configuration Tools** → **Standard Operating Procedures** → **Definition**.
2. Click the name of the standard operating procedure definition for which you want to launch an instance.
3. From the Definition Actions drop-down list, select **Launch**. A message is displayed that confirms that an instance of the standard operating procedure was created.

Figure 5-12 shows how to launch an SOP instance from the SOP configuration tool in the Solution Administration.

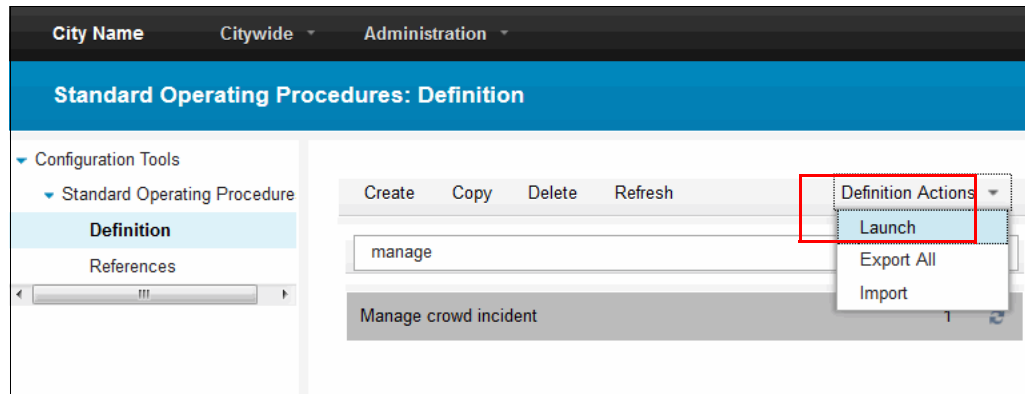


Figure 5-12 Launching an SOP instance from the Solution Administration

Launching an SOP instance from an event preview card

Users that are not authorized to the Standard operating procedures → Definition portal page can launch an SOP instance from an event preview card as long as they have the roles that are required to launch an SOP instance as described in Table 5-1 on page 114.

To launch an SOP instance from an event preview card, follow these steps:

1. Click the event to display the preview card.
2. Click **More Actions** → **Launch Standard Operating Procedure**.

Figure 5-13 on page 120 shows the More Actions menu in an event preview card.

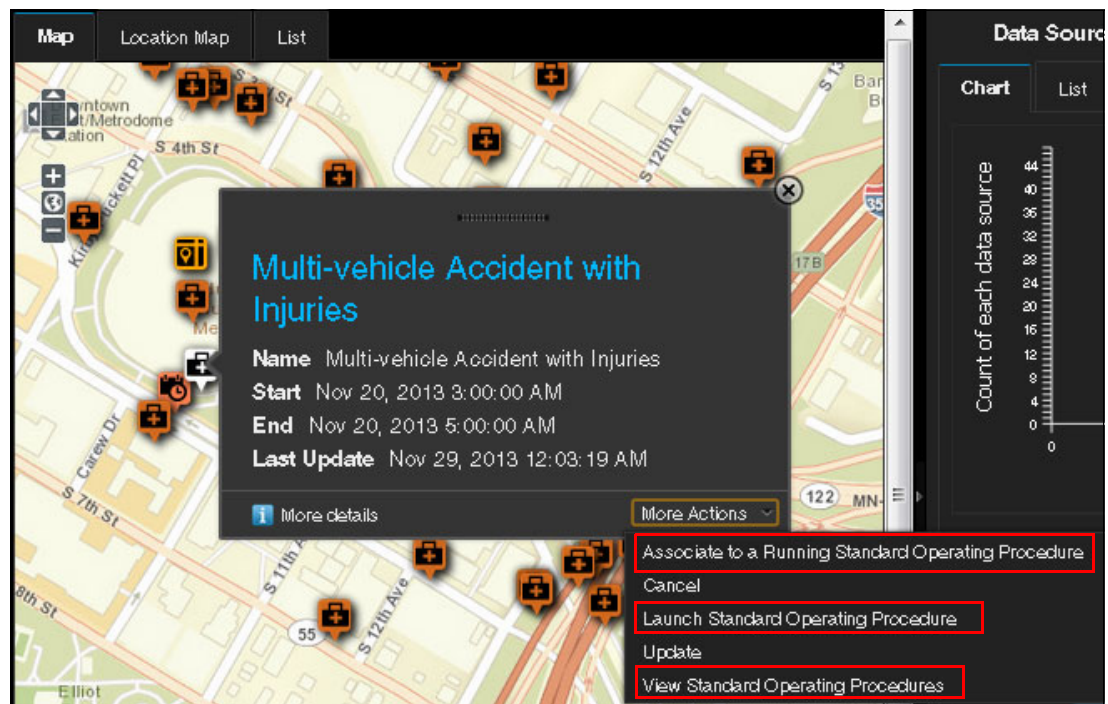


Figure 5-13 More Actions on events and incidents preview card

Figure 5-14 shows the window that opens when the operator clicks Associate to a Running Standard Operating Procedure in Figure 5-13. In this example, Request medical support is a running SOP instance.

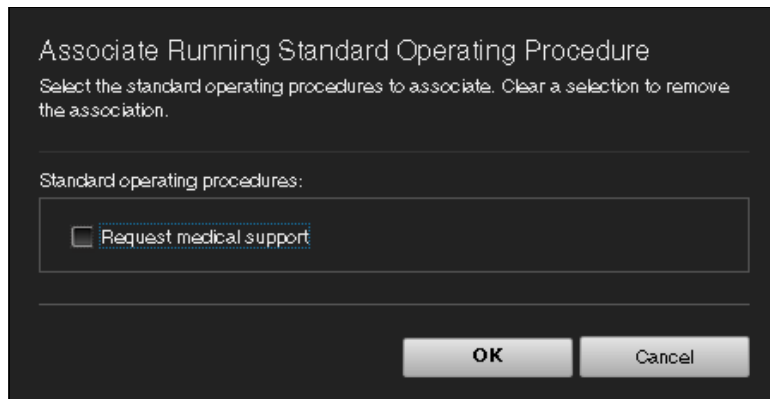


Figure 5-14 Associating running SOP to events or incidents

Figure 5-15 on page 121 shows the window that opens when the operator clicks Launch Standard Operating Procedure in Figure 5-13. The operator can select to view the SOPs associated with this type of event or incident by the data source routing expression or all approved SOP definitions.

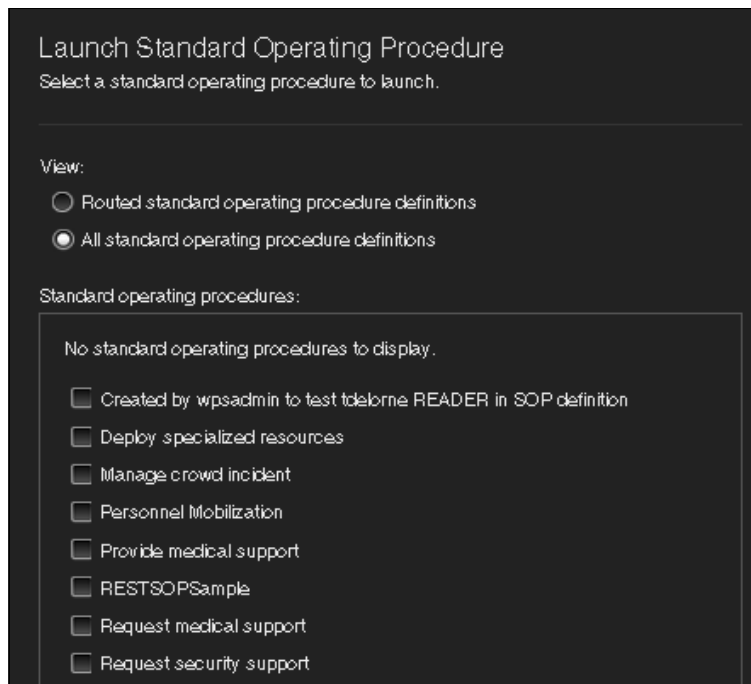


Figure 5-15 Launching SOP from operations view

Figure 5-16 shows the window that opens when the operator clicks View Standard Operating Procedures in Figure 5-13 on page 120.

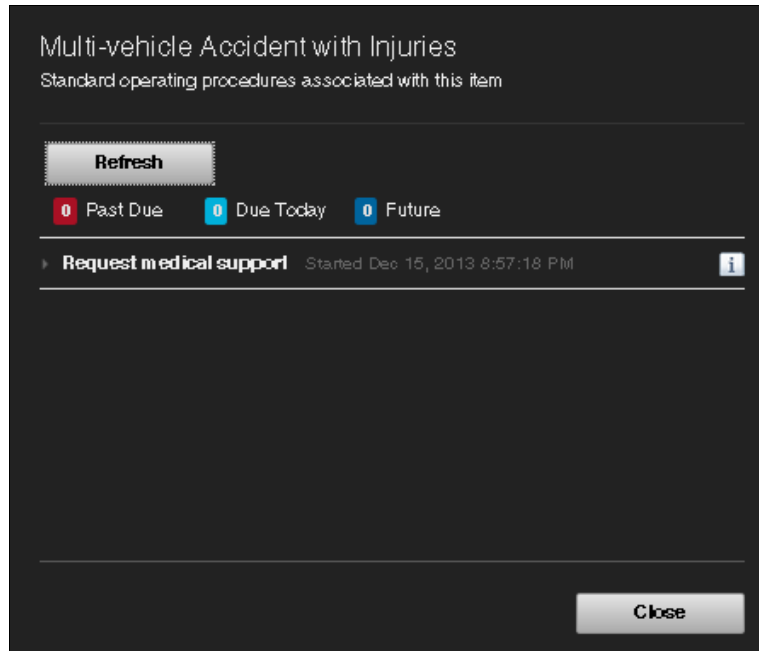


Figure 5-16 Viewing associated SOP instances

5.1.7 Viewing SOP instances

There are two ways to view an SOP instance:

- ▶ From the SOP configuration tool in the Solution Administration
- ▶ From the My Activities portlet

Viewing SOP instances from the Solution Administration

To view an SOP instance from the SOP configuration tool in the Solution Administration:

- ▶ The user must have one of the roles that are required to view an SOP instance as described in Table 5-1 on page 114:
 - User group is in the SopAdminRoles system property.
 - User group is in the Owners role in the SOP definition.
 - User group is in the Readers role in the SOP definition.
 - User group is in the Owners role in the SOP activities.
 - User group is in the Readers role in the SOP activities.
- ▶ The user is authorized to the Standard operating procedures → Definition portal page as described in “Granting access to the SOP Definition portal page” on page 111.

View the SOP instance from the Solution Administration:

1. In the Solution Administration view, click **Configuration Tools** → **Standard Operating Procedures** → **Definition**.
2. Click the name of the standard operating procedure definition for which you want to view the instance.
3. Click the **Instances** tab.
4. Click the SOP instance to view the details.

Figure 5-17 shows how to view the SOP instance from the Solution Administration.

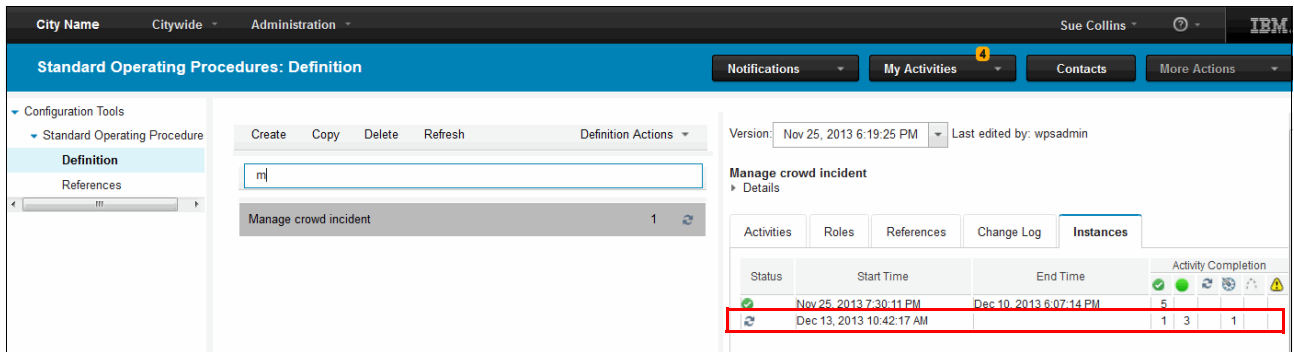


Figure 5-17 Viewing SOP instance from the Solution Administration

Figure 5-18 on page 123 shows the SOP instance details.

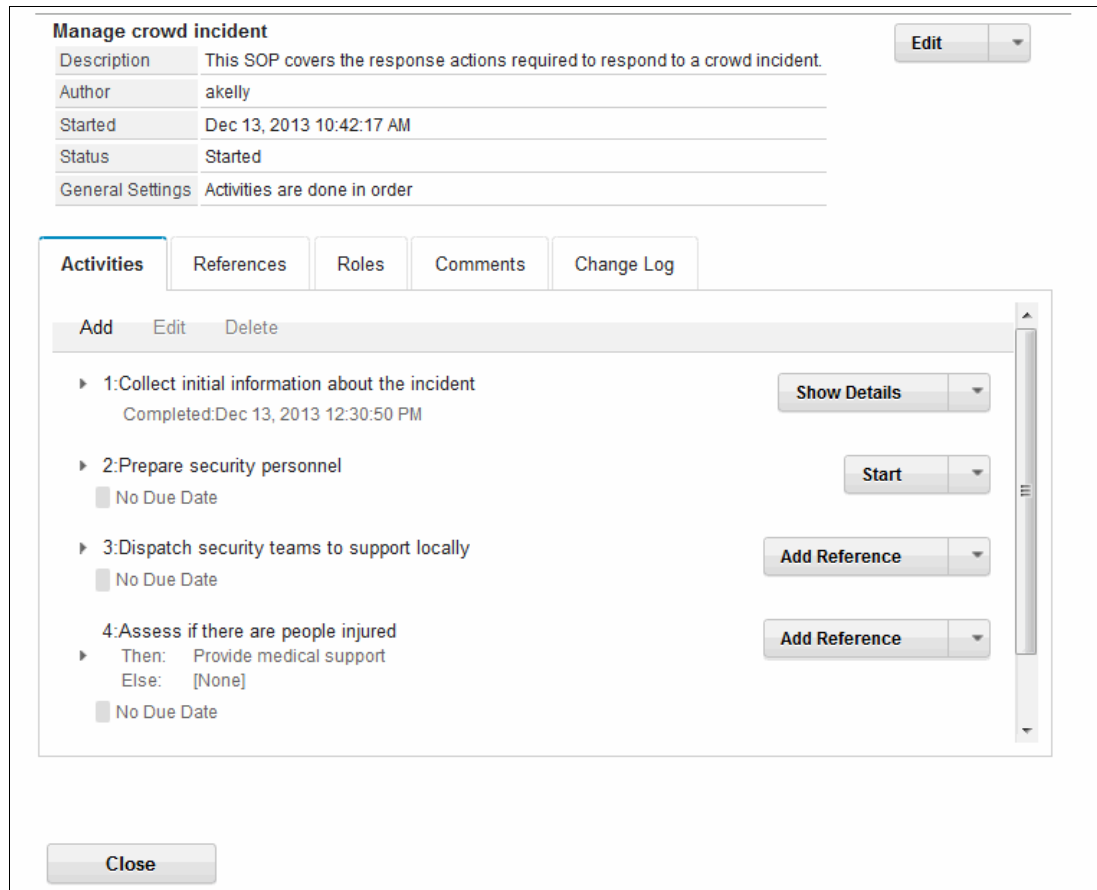


Figure 5-18 SOP instance details

Viewing SOP instance My Activities

Users that are not authorized to the Standard operating procedures → Definition portal page can view an SOP instance from My Activities as long as they have the roles required to view an SOP instance as described in Table 5-1 on page 114.

View an SOP instance from My Activities:

1. Click **My Activities**.
2. Click **More Actions** → **Launch Standard Operating Procedure**.

Figure 5-19 on page 124 shows how to view an SOP from My Activities.

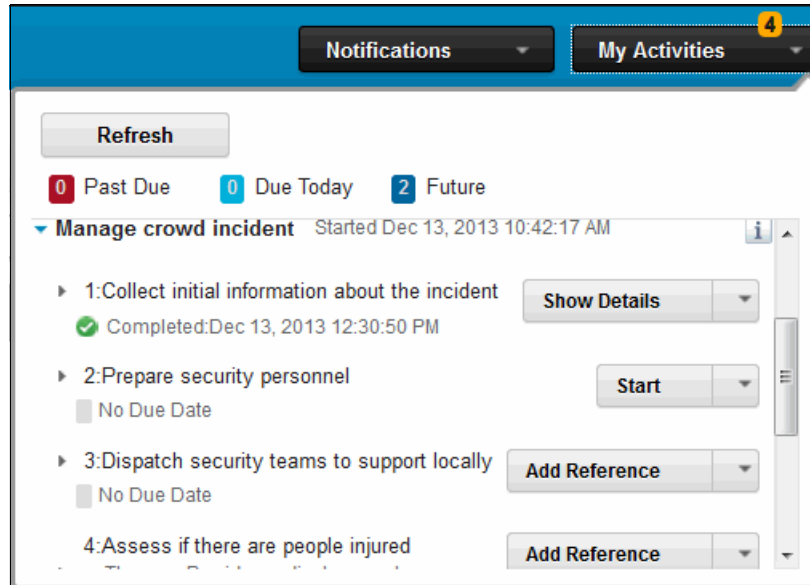


Figure 5-19 Viewing an SOP instance from My Activities

5.1.8 Reverting to a specific version

For approved SOPs, the complete list of its versions is in the Versions list, including the current draft version if there is one. Drafts can be created from any approved version of an SOP, but only one draft can exist at a time. To revert the SOP definition to an older version, it is necessary to create a draft of the target version and then approve it. The last approved version is the version of the SOP to be launched by the system. Follow these steps:

1. In the SOP definitions list, select the SOP definition.
2. From the Version list, select the older version to which you want to revert the SOP.
3. Click **Create Draft** and then **OK**.
4. Click **Save** to save the draft.
5. After the draft is saved, click **Submit for Approval**.
6. An authorized user can now select this SOP and then approve it. After it is approved, the SOP is ready to be launched. For more information about approving drafts, see 5.1.5, “Approving or disapproving drafts” on page 118.

5.1.9 Exporting and importing SOPs

You can export your SOP definitions as an XML file and import a previously exported SOP definitions file. This capability is useful in cases where the organization has predefined SOPs that were implemented using a different tool and also for migration purposes.

Export an SOP definition:

1. Select **Administration** → **Solution administration** → **Configuration Tools** → **Standard Operating Procedures** → **Definition**.
2. From the Definition Actions drop-down list, select **Export All**.
3. In the pop-up window that is displayed, click **Save File** → **OK**.
4. Navigate to the directory where you want to save the `SopDefinitions.xml` file and click **Save**.

Import an SOP definition from a previously exported XML file:

1. Select **Administration** → **Solution administration** → **Configuration Tools** → **Standard Operating Procedures** → **Definition**.
2. From the Definition Actions drop-down list, select **Import**.
3. In the Import Standard Operating Procedures window, choose the XML file to upload.

The imported definitions are displayed in the list of SOP definitions. You can import an SOP definition that has the same name as an existing SOP definition. In this case, the existing SOP definition is not overwritten. Instead, two SOP definitions with the same name are displayed in the list of definitions.

5.2 Configuring activities for different scenarios

One of the most important tasks for defining SOPs is to define the set of activities that form the procedure. The activities can be of different types and execution models. Any combination of different activities in an SOP is allowed.

When you define an SOP, you define activities that are included in the SOP.

To add an activity to an SOP definition, follow these steps:

1. In the Create Standard Operating Procedure Definition, select the **Activities** tab and click **Add**.
2. Enter the activity name and complete the optional fields as appropriate (see Figure 5-20).

Create Activity Definition

Required

Name *

Roles
 Owners:
 Readers:

Duration None

Description

Activity type

References

Name	Description
<input type="text"/>	<input type="text"/>

Figure 5-20 Creating an activity

When the SOP is launched, an instance of each activity definition that is part of the SOP is created. How the activity starts is defined by the activity type.

For each instance of an SOP that has been launched, the My Activities window displays all the activities, including those activities that are complete.

Figure 5-21 shows the Managed crowd incident SOP that has been launched and the instances of the activities that are part of the SOP.

Figure 5-21 Activities instances in a launched SOP - My Activities window

The activities for the instance of an SOP that has been launched can have one the following statuses: active, waiting, started, skipped, auto started, auto completed, or completed. During the execution of an SOP, after all required activities are complete, the SOP is complete.

All activity management functions are on the Activities tab, under the Standard Operating Procedures Definition view.

It is important to understand the concept of activity types and how to specify the sequence in which activities need to be executed before configuring SOPs.

The following configuration attributes affect how the activities are executed:

- ▶ *Activities are done in order*: This attribute is specified in the SOP definition. It indicates that the activities in the SOP must be executed sequentially.
- ▶ *Required*: This attribute is specified for the activity. If the activity is required, it cannot be skipped.
- ▶ *Activity type*: Determines special characteristics of the activity, for example, how the activity is triggered or started.

Table 5-2 on page 127 lists the available activity types and their description.

Table 5-2 SOP activity types

Activity type	Description
Manual Activity	An activity that is done manually by the owner. Provide details in the Description field.
Automation Activity	An activity that initiates and tracks a particular work order in IBM SmartCloud Control Desk 7.5.1. Select a predefined work order from the list.
Notification Activity	An activity that displays a notification window that contains an email template for the activity owner to complete, and then sends an email notification. Notification templates are stored as references in the reference library. To select the email template that will be displayed to the owner, select a reference from Reference name. To preview the email template, click Preview.
If-Then-Else Activity	A conditional activity that allows branching based on specific criteria. Either enter or select values for Then and Else .
Representational State Transfer (REST) Service Activity	An activity that creates a REST service call: <ul style="list-style-type: none"> ▶ For REST service, enter or select a REST service URL. ▶ For USER ID, enter a user ID to access the REST service. ▶ For Password, enter a password to access the REST service.
SOP Activity	An activity that launches another standard operating procedure. From the SOP list, select the name of a standard operating procedure.

5.2.1 Configuring activities to execute sequentially

The activities of an SOP are executed sequentially when *Activities are done in order* is specified in the General Settings section of the SOP definition (see Figure 5-22 on page 128).

Note: *Activities are done in order* applies to all activities within the SOP.

If it is specified that activities are done in order, you will not be able to start an activity until the predecessor activities that it depends on are completed.

Specify that activities must be executed sequentially for workflows where the activities must be executed in a chronological order or where activities depend on the result of previous activity.

Example of sequential activities:

- ▶ MA1 - Approve operation
- ▶ SA2 - Collect information about incident area and number of victims
- ▶ SA3 - Prepare personnel for standby
- ▶ SA4 - Redirect traffic to clear incident area

Figure 5-22 on page 128 shows an activity flow with activities that run in sequential order.

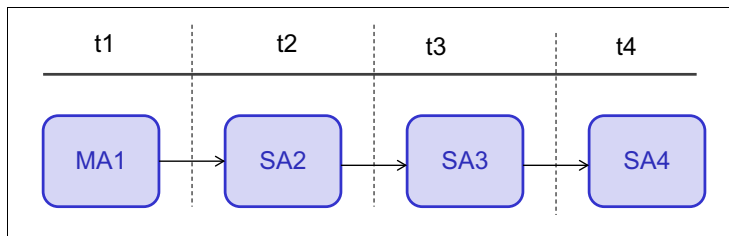


Figure 5-22 Sequential activity flow

Notes:

- ▶ The user can start SA2 only after MA1 completes, SA3 after SA2 completes, and SA4 after SA3 completes. The sequential property of an SOP applies to all activities (it is either all activities are sequential or none for the entire SOP).
- ▶ The variables *t1*, *t2*, *t3*, and *t4* represent the execution time.

To define an SOP so that its activities are executed in a sequential order, on the Standard Operating Procedure Definition Basics tab, check **Activities are done in order** in the General Settings section (see Figure 5-23).

Version: Draft Last edited by: wpsadmin Submit for Approval

Name * Response to Event Save Cancel

Description Description for Response to Event

General Settings Activities are done in order

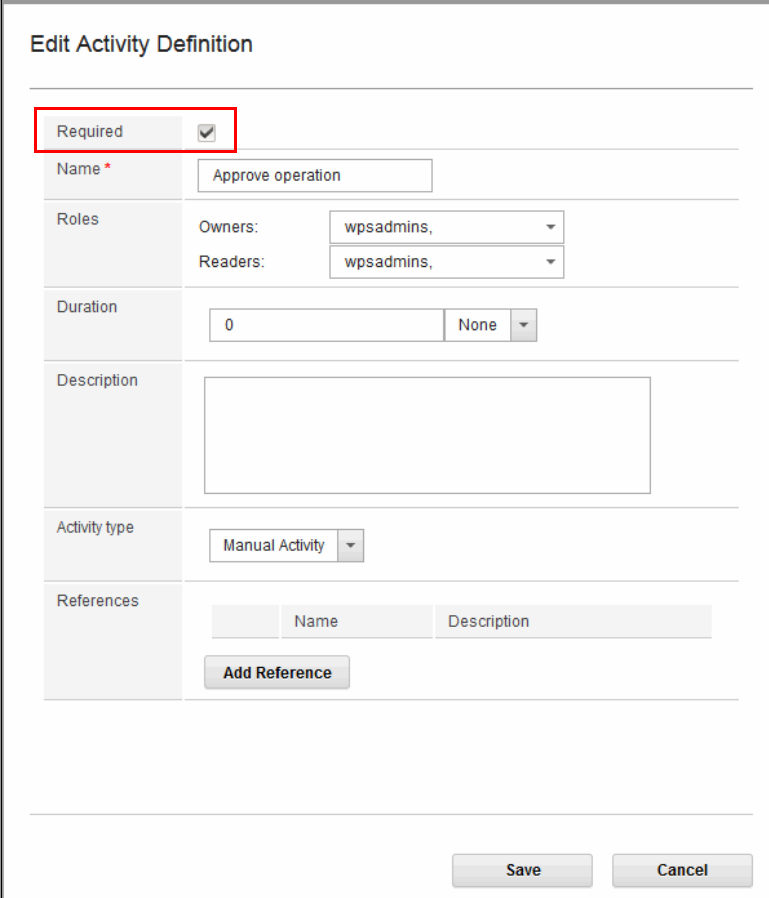
Figure 5-23 SOP definition - Activities are done in order

5.2.2 Configuring required activities

A *required activity* is mandatory to the SOP. You cannot skip an activity that is specified as a required activity in the SOP definition.

To specify that the activity is required, check the field **Required** in the Create Activity Definition or Edit Activity Definition window. Any of the activity types can be configured as *required* (see Figure 5-24 on page 129).

Note: Required is an attribute of an activity. SOPs can have activities that are required and activities that are not required (they can be skipped).



The screenshot shows the 'Edit Activity Definition' window. The 'Required' checkbox is checked and highlighted with a red box. The activity name is 'Approve operation'. The roles are 'wpsadmins' for both Owners and Readers. The duration is '0' and the activity type is 'Manual Activity'. There is an 'Add Reference' button and 'Save' and 'Cancel' buttons at the bottom.

Edit Activity Definition			
Required	<input checked="" type="checkbox"/>		
Name *	Approve operation		
Roles	Owners: wpsadmins, Readers: wpsadmins,		
Duration	0 None		
Description			
Activity type	Manual Activity		
References	<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody></tbody></table> Add Reference	Name	Description
Name	Description		
Save Cancel			

Figure 5-24 Configuring a required activity

The following example shows a required activity (bold):

- ▶ **MA1 - Approve operation**
- ▶ A2 - Collect information about incident area and number of victims
- ▶ A3 - Prepare personnel for standby
- ▶ A4 - Redirect traffic to clear incident area

Figure 5-25 on page 130 shows an activity flow with a required activity.

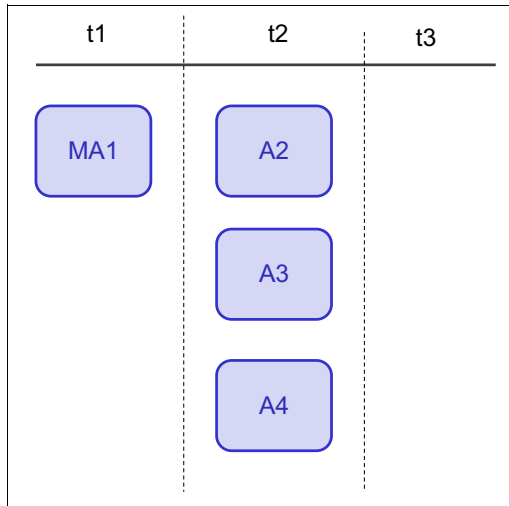


Figure 5-25 Required activities flow

Note: The variables $t1$, $t2$, and $t3$ represent the execution time. Except MA1, all the other activities are optional and do not need to be performed sequentially. A2, A3, and A4 can be run in parallel.

5.2.3 Configuring manual activities

Manual activity type is an activity that is performed manually. Manual activities are the most basic and essential type. This activity type must be manually performed by the owner, after the SOP is launched.

To configure a manual activity, select **Manual Activity** in the Activity type list in the Create Activity Definition or Edit Activity Definition window (see Figure 5-26 on page 131).

Create Activity Definition

Required

Name *

Roles
 Owners:
 Readers:

Duration None

Description

Activity type

References

Name	Description
<input type="text"/>	<input type="text"/>

Figure 5-26 Configuring a manual activity

The following examples are manual activities:

- ▶ A1 - Collect information about the incident area and number of victims
- ▶ A2 - Prepare personnel for standby
- ▶ A3 - Redirect traffic to clear incident area

Figure 5-27 shows an activity flow with activities that are started manually.

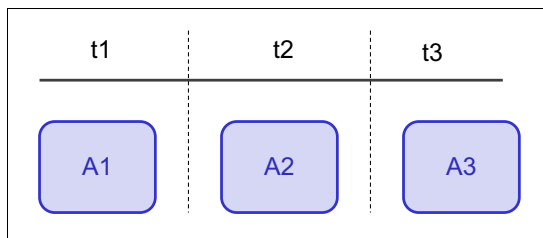


Figure 5-27 Manual activities flow

Note: The variables $t1$, $t2$, and $t3$ represent the execution time.

5.2.4 Configuring automation activities

The SOP configuration tool in IBM Intelligent Operations Center enables city officials to configure SOP definitions. If the procedures require more complex workflows, an automation activity can be configured to trigger a work order in IBM SmartCloud Control Desk.

IBM SmartCloud Control Desk job plans can handle a large deployment of resources. For instance, establishing an emergency center might require a large mobilization of resources (the buildings must be staffed, power turned on, supplies and furniture obtained, and so on).

An *automation activity* initiates and tracks a particular work order in IBM SmartCloud Control Desk. The work order must be predefined in the IBM SmartCloud Control Desk so that it appears in the list when this type of activity is selected.

Work orders are based on job plans that contain a detailed description of work, such as information regarding estimated human resources, skills, hours, materials, services, and required tools to perform the tasks. For more information, see *Job plans overview* at this website:

<http://bit.ly/1mJLZRI>

To configure an automation activity, select **Automation Activity** in the Activity type list in the Create Activity Definition or Edit Activity Definition window (see Figure 5-28 on page 133).

Edit Activity Definition

Required	<input type="checkbox"/>				
Name *	Put personnel on standby				
Roles	Owners: CityWideExecutive, Readers: CityWideSupervisor,				
Duration	0 None				
Description					
Activity type	Automation Activity Job plan: Prepare personnel for standby (Prepare)				
References	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">Add Reference</td> </tr> </tbody> </table>	Name	Description	Add Reference	
Name	Description				
Add Reference					

Save Cancel

Figure 5-28 Activity definition window with Activity type set to Automation Activity

Figure 5-29 shows a job plan that is executed by the automation activity SA3.

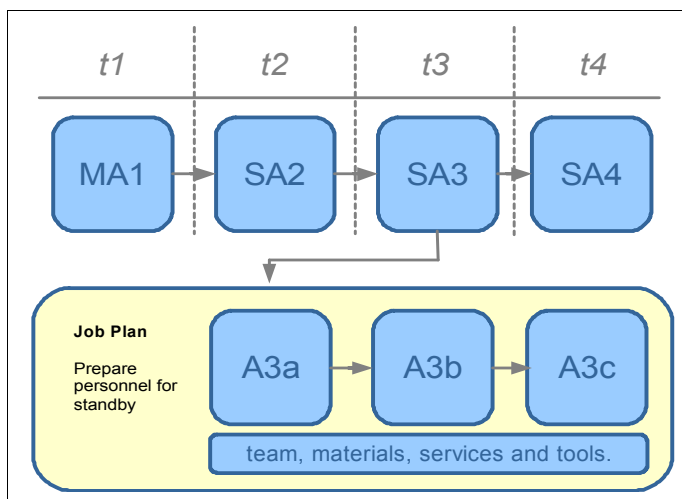


Figure 5-29 Automation flow showing automation activity triggering a job plan

Note: When SA3 runs, it automatically triggers the job plan Prepare personnel for standby.

5.2.5 Configuring If-Then-Else activities

If-Then-Else activities are conditional activities that allow branching based on specific criteria. When the activity is launched, the operator decides whether to execute a set of actions (Then) or another (Else).

The If-Then-Else activity is useful in situations where you want to use the results from other activities to make a decision and launch a new SOP to respond to the situation.

The following examples show conditional activity:

- ▶ MA1 - Approve operation
- ▶ SA2 - Collect information about incident area and number of victims
- ▶ SA3 - Prepare personnel for standby:
 - SA3 If team is not available = true → Then, initiate contingency plan
- ▶ SA4 - Redirect traffic to clear incident area

Figure 5-30 shows a conditional activity flow.

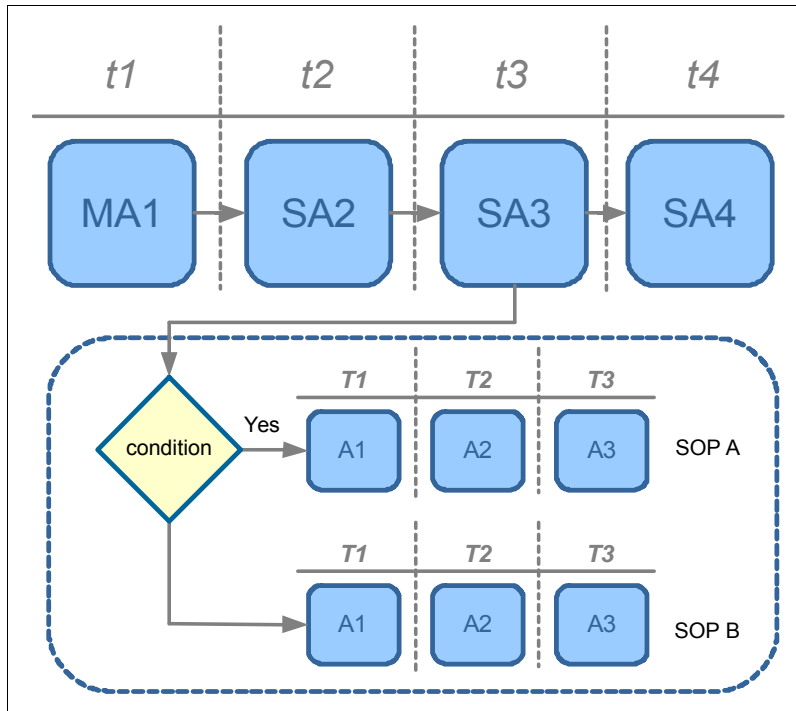


Figure 5-30 Conditional activity flow

Create an If-Then-Else Activity by selecting this type in the Activity type selection drop-down list box. After this activity type is selected, two other selection fields are displayed: one for Then and the other one for Else. A list of approved SOPs is displayed. It is required to select one SOP for the Then branch. The Else branch is optional (see Figure 5-31 on page 135).

Create Activity Definition

Required	<input type="checkbox"/>
Name *	Assess if there are people injured
Roles	Owners: <input type="text"/> Readers: <input type="text"/>
Duration	0 <input type="text"/> None
Description	Request medical services if there are people injured; if not, request security team support.
Activity type	If-Then-Else Activity Then: Request medical support Else: Request security support

Figure 5-31 If-Then-Else Activity definition

Figure 5-32 shows selecting the Request medical support SOP for the Then branch.

Activity type	If-Then-Else Activity
Then:	Request medical support
Else:	Deploy specialized resources Manage crowd incident Personnel Mobilization Provide medical support Request medical support
References	

Figure 5-32 Then SOP

When the SOP instance runs, the operator decides whether to launch the Then SOP, or the Else SOP if one is available, by selecting the Start Then option or the Start Else option as shown in Figure 5-33 on page 136.

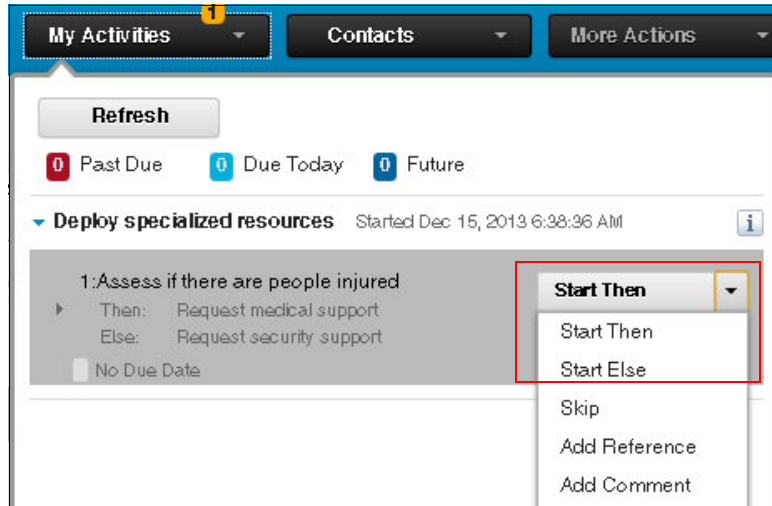


Figure 5-33 Running If-Then-Else activity

Figure 5-34 shows the Request security support SOP that is launched when the operator selects the Else option.

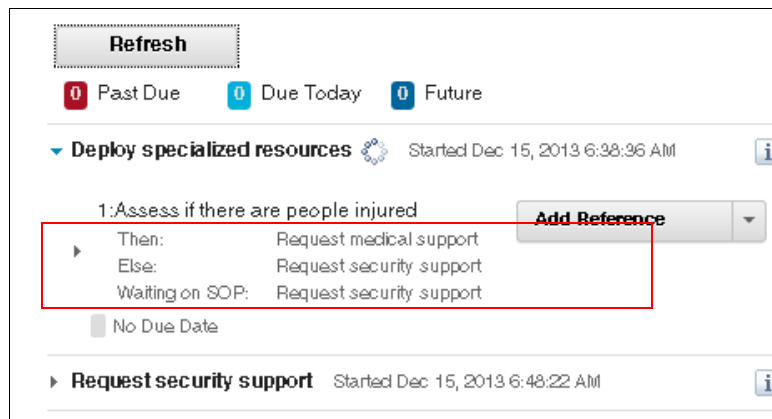


Figure 5-34 Request security support SOP started from Deploy specialized resources SOP

5.2.6 Configuring notification activities

Notification activities enable the IBM Intelligent Operations Center operator to complete an email template and send an email notification when the activity starts.

It is required that you set up the Simple Mail Transfer Protocol (SMTP) server that will send the email notifications before configuring and launching a notification activity.

To set up the SMTP, update the following system properties:

- ▶ MailServerHostname: Host name of the SMTP server
- ▶ MailServerPort: SMTP server port
- ▶ MailSender: Sender of emails to be sent by the activity. It appears in the From field.

To update the system properties, from the Solution Administration, expand **System Administration** and click **System Properties** as shown in Figure 5-35 on page 137.

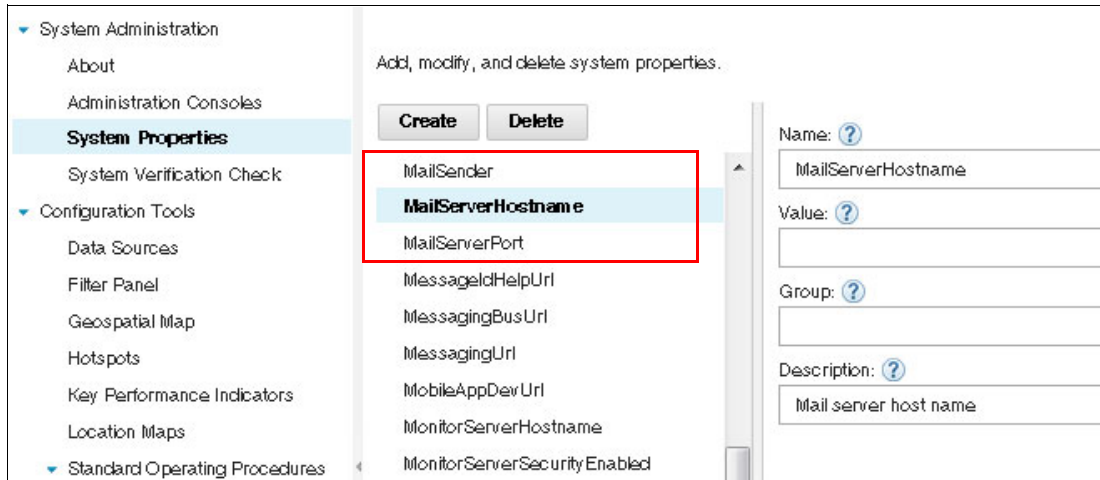


Figure 5-35 Mail server system properties

IBM Intelligent Operations Center uses the mail server system properties to access the simultaneous multithreading (SMT) server that delivers the email messages.

Email templates are created in IBM Intelligent Operations Center and reused in notification activities. Email templates are stored as *references*.

To create a notification template, follow these steps:

1. Create a text file that contains the email template. Example 5-1 shows a template example. The To, Subject, and Body fields are all optional.

Example 5-1 Notification template

To:
 Subject: Attention: Emergency incident detected
 Body: Emergency incident detected. Please contact Operations Center for further information.

2. Publish the template in a URL accessible from IBM Intelligent Operations Center.
3. Create a reference by selecting **Solution Administration** → **Standard Operating Procedures** → **References** and pointing to the URL where the template is published (Figure 5-36 on page 138).

Add a Reference

Name * Notification template

URI * http://test-url/notification

Description NOTIFICATION

Private

Add Cancel

Figure 5-36 Adding a reference in the reference library pointing to a URL containing an email template

Note: Ensure that the first line of the description contains only the keyword NOTIFICATION. You can add further description details in subsequent lines.

To create a notification activity, follow these steps:

1. Create an SOP definition as described in 5.1.2, “Configuring SOP definitions” on page 115.
2. Add an activity type **Notification Activity** as shown in Figure 5-37 on page 139:
 - a. In the Reference name pull-down list, select the email **Notification Template**.
 - b. Click **Preview** to test access to the template.

Create Activity Definition

Required	<input type="checkbox"/>				
Name *	Notify medical service				
Roles	Owners: <input type="text"/> Readers: <input type="text"/>				
Duration	0 None				
Description	<input type="text"/>				
Activity type	Notification Activity				
Reference name	Notification template Preview				
References	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">Add Reference</td> </tr> </tbody> </table>	Name	Description	Add Reference	
Name	Description				
Add Reference					

Figure 5-37 Creating a notification activity

During the execution of the SOP, when the operator starts a notification activity, a Notification form is displayed with the notification template associated with the activity as shown in Figure 5-38 on page 140.

Notification

Notification template:	Notification template
To: *	medicals@city.org
Subject: *	Attention: Emergency incident detected
Message:	<p>Emergency incident detected. Please contact Operations Center for further information.</p> <p>Activity: Notify medical service Description Owner: undefined Status: ACT_WAITING</p> <p>SOP: Request medical support Description UNKNOWN Owner: undefined Status: SOP_STARTED</p> <p>There are no events associated with this standard operating procedure.</p>
Related to:	Activity <input type="button" value="Append"/>
User (optional)	
Password (optional)	

Figure 5-38 Notification form displayed when the notification activity starts

The operator can update the form by providing the following information:

- ▶ To: Email address or the IBM Intelligent Operations Center user to whom the email will be sent.
- ▶ Subject: Message subject.
- ▶ Message: Content of the message.
- ▶ Related to: Using this field, the operator can specify to append additional information to the message related to the following areas (see Figure 5-39 on page 141):
 - Activity: Information about the notification activity
 - SOP: Information about the SOP
 - Event: Information about the event associated with this SOP

Message:	<p>Emergency incident detected. Please contact Operations Center for further information.</p> <p>Activity: Notify medical service Description Owner: undefined Status: ACT_WAITING</p> <p>SOP: Request medical support Description UNKNOWN Owner: undefined Status: SOP_STARTED</p> <p>There are no events associated with this standard operating procedure.</p>	
Related to:	Activity	Append
User (optional)	Activity	
Password (optional)	SOP	
	Event	

Figure 5-39 Appending information to the notification message

5.2.7 Configuring REST service activities

This type of activity creates a REST service call to access an external entity. In the activity definition, for REST service, enter or select a REST service URL. You must also specify a user ID to access the REST service and a password for that user ID.

The following example shows a REST activity (bold):

- ▶ MA1 - Approve operation
- ▶ SA2 - Collect information about incident area and number of victims
- ▶ SA3 - Prepare personnel for standby
- ▶ **SA4 - Send voice message to tactic team**

Figure 5-40 shows a REST activity flow.

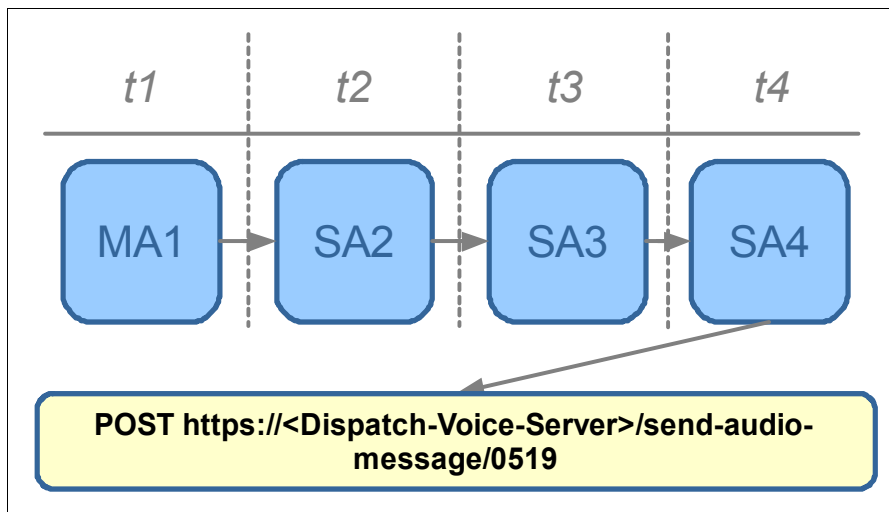


Figure 5-40 REST activity flow

Note: The REST service activity can also be used to connect an SOP to another SOP running on a different IBM Intelligent Operations Center instance to execute tasks, such as launch an SOP or start an activity.

Figure 5-41 shows the Create Activity Definition window with the REST Service activity type selected.

The screenshot shows the 'Create Activity Definition' window. The 'Activity type' section is highlighted with a red box. It contains the following fields:

- Activity type:** REST Service
- REST Service:** https://iop20lhs.ibmplatform
- User ID:** wpsadmin
- Password:** masked with asterisks

Other fields in the window include:

- Required:** unchecked checkbox
- Name:** Allocate medical service
- Roles:** Owners and Readers dropdowns
- Duration:** None
- Description:** This activity launches an IOC SOP via REST service
- References:** Table with columns Name and Description, and an Add Reference button.
- Buttons:** Add and Cancel at the bottom.

Figure 5-41 REST activity definition

When the REST Service activity type is selected, the following fields are enabled:

- ▶ **REST Service:** Target REST service to be invoked by this activity. An IBM Intelligent Operations Center REST service URL can be provided if the target REST service is an IBM Intelligent Operations Center REST service.

Figure 5-41 shows an example of using a REST service to launch an SOP in a remote IBM Intelligent Operations Center instance. The REST service URL in this example is `taps://<remote_ioc_ibs>/ioc/api/sop-service/sop?sopDefinitionId=665`.

For more information about launching SOPs via a REST service, see 5.3.4, “Working with SOP instances” on page 148.

- ▶ **User ID:** User with permissions to launch the target REST service. This example is an IBM Intelligent Operations Center user.
- ▶ **Password:** User’s password.

When the SOP is launched and the REST service activity starts, it invokes the target REST service. In this example, it starts the SOP with definition ID 665 (Request medical support SOP) in the target IBM Intelligent Operations Center server, as shown in Figure 5-42.

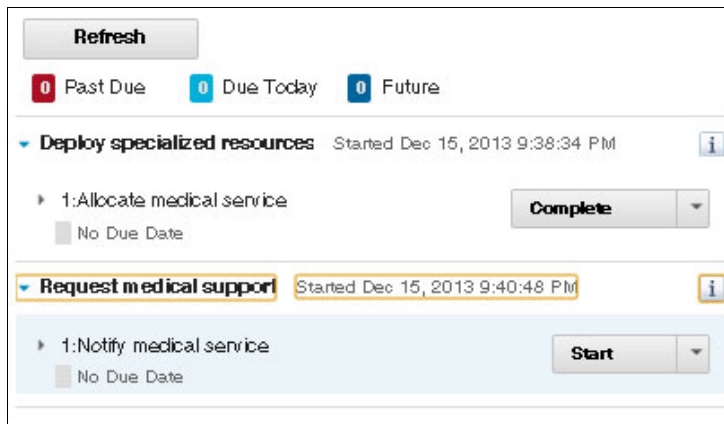


Figure 5-42 Request medical support SOP started from a REST activity

5.2.8 Configuring SOP activities

An SOP can call another SOP during its execution. The calling SOP (parent) must include an activity of the type SOP. In the Activity Definition view, select the SOP Activity type and then, from the SOP list, select the name of the subordinate (child) SOP to call when the activity is instantiated.

The following shows an SOP activity:

- ▶ MA1 - Approve operation
- ▶ SA2 - Collect information about incident area and number of victims
- ▶ SA3 - Prepare personnel for standby
- ▶ **SA4 - Isolate impacted area**

Figure 5-43 shows an SOP activity flow.

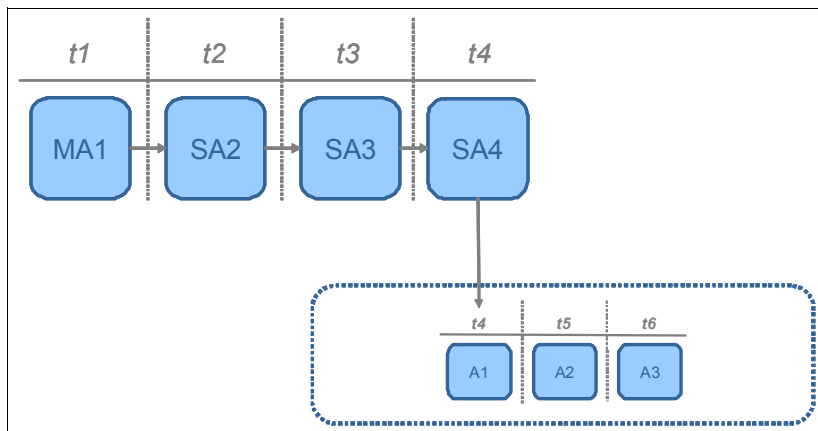


Figure 5-43 SOP activity flow

Note: SA4 starts the child SOP in the current time slide (t4). An activity that starts a child SOP cannot be completed until the child SOP is completed (the completion of the child SOP will auto complete the activity in the parent SOP).

5.3 SOP REST service

IBM Intelligent Operations Center provides a service interface that enables developers to extend the solution programmatically. This section provides an overview of the Representational State Transfer (REST) services available for SOPs. IBM Intelligent Operations Center also provides a server-side Java interface (not described in this chapter). For more information, see Chapter 1, “IBM Intelligent Operations Center programming model overview” on page 1.

The SOP REST service provides access to SOP resources within IBM Intelligent Operations Center.

The SOP REST service provides a set of uniform resource identifiers (URIs). For each method, the HTTP request consists of the following components:

- ▶ Host and port information
- ▶ Resource URI
- ▶ Optional parameters

The base URI for the SOP REST Service is `/ibm/ioc/api/sop-service/`.

To access these services, a programmer must develop RESTful applications that use HTTP requests to query and update SOP data. Use HTTP basic access authentication with an IBM Intelligent Operations Center user with the required authority to call the REST services.

The requests and response are in JavaScript Object Notation (JSON) format. Use MIME type: `application/json` to parse the IBM Intelligent Operations Center REST service responses in a REST application. Any REST client can be used to test the services described in this section.

This section includes examples that show how to access some of the SOP REST services.

Note: To test the examples in the following sections, one of the generally available REST clients was used.

5.3.1 Retrieving all SOP definitions

This REST service is one of the SOP REST services most frequently used because it retrieves all SOP approved definitions. This service is usually the starting point to access other SOP REST services.

Example 5-2 shows the request to access this service via the GET method. This request requires no parameters.

Example 5-2 Retrieving all SOP definitions

```
https://<ioc_ihs>/ibm/ioc/api/sop-service/sopDefinition
```

This request returns a JSON object and includes the list of all SOP definitions as shown in Example 5-3 on page 144. Only the most recently approved definitions are returned, not draft versions. Use the returned SOP definition ID (uid) to query specific SOPs.

Example 5-3 Response sample - all SOP definitions (only one SOP definition shown in the example)

```
"matchAll": false,  
"trigAutomatic": false,
```

```

"uid": 106,
"cancelEvent": false,
"archived": false,
"author": "wpsadmin",
"description": "The Manage Event SOP covers all steps required to prepare for and
run the event at City A.",
"name": "Manage event",
"ordered": true,
"statusString": "SOP_APPROVED",
"status": 3,
"transitional": false,
"version": 1384430064761,
"versionUid": 105,
"sop": {
  "$ref": "/sopDefinition/106/sop"
},
"versions": {
  "$ref": "/sopDefinition/106/versions"
},
"instance": {
  "$ref": "/sopDefinition/106/instance"
},
"changeLog": {
  "$ref": "/sopDefinition/106/changeLog"
},
"triggerParameter": {
  "$ref": "/sopDefinition/106/triggerParameter"
},
"reference": {
  "$ref": "/sopDefinition/106/reference"
},
"roleAccess": {
  "$ref": "/sopDefinition/106/roleAccess"
},
"activityDefinition": {
  "$ref": "/sopDefinition/106/activityDefinition"
},
"versionString": "Nov 14, 2013 11:54:24 AM",
"activeInstanceCount": 0,
"accessType": "ACCESS_READ_WRITE",
"actions": [{
  "actionName": "action.addReference",
  "actionUri": "sopDefinition/106/reference"
},
{
  "actionName": "action.createDraft",
  "actionUri": "sopDefinition/106/createDraft"
}]

```

Each SOP definition is delimited by “{” and “}”, between the matchAll and actions properties. Table 5-3 shows the main properties returned in the response for each SOP definition. For the complete list of SOP definition properties, see *IBM Intelligent Operations Center V1.6 REST APIs*.

Table 5-3 Subset of SOP definition properties

Property	Type	Description
uid	Integer	SOP definition ID
author	String	SOP creator
description	String	SOP description
name	String	SOP name
ordered	Boolean	Whether the activities in this SOP must be executed in order
statusString	String	Status of the SOP definition. Possible values: SOP_APPROVED SOP_DRAFT SOP_PENDING_APPROVAL SOP_UNKNOWN

The response shown in Example 5-3 on page 144 includes REST endpoints that can be used to query more information about each SOP. These endpoints are indicated by \$ref in the response and can all be accessed with a GET request and the following URIs:

- ▶ /sopDefinition/<SOP_DEFINITION_ID>/sop returns the definition of an SOP with an ID of <SOP_DEFINITION_ID>.
- ▶ /sopDefinition/<SOP_DEFINITION_ID>/versions returns the approved versions of an SOP with an ID of <SOP_DEFINITION_ID> in descending order.
- ▶ /sopDefinition/<SOP_DEFINITION_ID>/instance returns the instances associated with an SOP with an ID of <SOP_DEFINITION_ID>.
- ▶ /sopDefinition/<SOP_DEFINITION_ID>/changeLog returns a log of changes to the SOP, as shown in Example 5-4.

Example 5-4 SOP change log response

```
uid: 676
attribute: "status"
changeTs: 1386955862648
from: "SOP_DRAFT"
to: "SOP_PENDING_APPROVAL"
user: "wpsadmin"
actualStartTsString: "13/12/2013 17:31:02"
```

- ▶ /sopDefinition/<SOP_ID>/triggerParameter is used to trigger a parameter of this SOP.
- ▶ /sopDefinition/<SOP_DEFINITION_ID>/reference returns references associated to the SOP.
- ▶ /sopDefinition/<SOP_DEFINITION_ID>/roleAccess returns roles and access for the SOP.
- ▶ /sopDefinition/<SOP_DEFINITION_ID>/activityDefinition returns the list of activity definitions associated with the SOP.

5.3.2 Retrieving an SOP definition

To retrieve a specific SOP definition, first it is necessary to query all SOPs, as described in 5.3.1, “Retrieving all SOP definitions” on page 144, to retrieve the uid of the SOP to query.

Example 5-5 shows the request to access this service via the GET method. The ID of the SOP is sent as a parameter in the URL.

Example 5-5 Retrieving an SOP definition

```
https://<ioc_ihs>/ibm/ioc/api/sop-service/sopDefinition/<SOP_DEFINITION_ID>
```

The response contains an SOP definition as explained in Example 5-3 on page 144 and Table 5-3 on page 146.

5.3.3 Retrieving activity definitions

Retrieving an SOP activity definition is also possible after querying an SOP ID.

Example 5-6 shows a request to retrieve activity definitions for an SOP with an ID of <SOP_ID> via the GET method. The SOP ID is sent as a parameter in the URL.

Example 5-6 Retrieving the activity definition list of an SOP

```
https://<ioc_ihs>/ibm/ioc/api/sop-service/sopDefinition/<SOP_DEFINITION_ID>/activityDefinition
```

The response includes the list of all activity definitions as shown in Example 5-7.

Example 5-7 Response sample - Activity definitions of an SOP (only one activity definition shown)

```
[{
  "uid": 108,
  "required": true,
  "actOrder": 1,
  "archived": false,
  "description": "",
  "durationUnitString": "durationNone",
  "durationUnit": 0,
  "estDuration": 0.0,
  "name": "Allocate personnel and resources",
  "transitional": false,
  "versionUid": 0,
  "activities": {
    "$ref": "/activityDefinition/108/activity"
  },
  "changeLog": {
    "$ref": "/activityDefinition/108/changeLog"
  },
  "activityDefinitionProperty": {
    "$ref": "/activityDefinition/108/activityDefinitionProperty"
  },
  "reference": {
    "$ref": "/activityDefinition/108/reference"
  },
  "sopDefinition": {
    "$ref": "/sopDefinition/108"
  }
}]
```

```

    },
    "roleAccess": {
      "$ref": "/activityDefinition/108/roleAccess"
    },
    "accessType": "ACCESS_READ_WRITE"
  },
  ...
}]

```

Each activity definition is delimited by “{” and “}”, between the uid and the accessType properties. Table 5-4 shows the main properties returned in the response for each activity definition.

Table 5-4 Subset of activity definition properties

Property	Type	Description
uid	Integer	Activity definition ID
required	Boolean	Whether this activity is required for SOP completion
actOrder	Integer	Order in which this activity must be completed relative to other activities in the SOP
description	String	Activity description
durationUnit	Integer	Activity duration unit. Possible values are 0 for none, 1 for minutes, 2 for hours, or 3 for days
estDuration	Integer	Activity duration
name	String	Activity name

5.3.4 Working with SOP instances

SOPs can be launched via a REST interface. A REST application must use the HTTP POST method to access this service. Example 5-8 shows the request to launch an SOP.

Example 5-8 Request to launch an SOP instance

https://<ioc_ihs>/ibm/ioc/api/sop-service/sop?sopDefinitionId=<SOP_DEFINITION_ID>

POST Request content:

```

{
  "DATASOURCEID":0,
  "OBJECTID":0
}

```

The ID of the SOP definition is sent as a parameter. The SOP definition must be in the approved status.

This request returns the properties of the started SOP, that is, the newly created instance of the SOP as shown in Example 5-9 on page 149.

```
{
  "uid": 597,
  "cancel_Event": false,
  "archived": false,
  "description": "test",
  "name": "Launch Test Def",
  "ordered": false,
  "statusString": "SOP_STARTED",
  "status": 2,
  "activity": {
    "$ref": "/sop/597/activity"
  },
  "sopDefinition": {
    "$ref": "/sopdefinition/584"
  },
  "changeLog": {
    "$ref": "/sop/597/changeLog"
  },
  "comment": {
    "$ref": "/sop/597/comment"
  },
  "reference": {
    "$ref": "/sop/597/reference"
  },
  "roleAccess": {
    "$ref": "/sop/597/roleAccess"
  },
  "creator": "wpsadmin",
  "actualStartTs": 1381245875661,
  "accessType": "ACCESS_NONE",
  "actions": [
    {
      "actionName": "action.addComment",
      "actionUri": "/sop/597/comment"
    }
  ],
  "event": {
    "$ref": "/sop/597/event"
  }
}
```

A subset of the SOP instance properties is listed in Table 5-5 on page 150.

Table 5-5 Subset of SOP instance properties

Property	Type	Description
uid	Integer	ID of the SOP instance
sopDefinition	Object	Relative URL to the SOP Definition related to this instance: /sopDefinition/<SOP_ID>
actualStartTs	Integer	Start time of this SOP instance
comment	Object	Comment in the SOP details

Example 5-10 shows a request to retrieve all instances of an SOP via the GET method.

Example 5-10 Retrieving all SOP instances

```
https://<ioc_ihs>/ibm/ioc/api/sop-service/sop
```

5.3.5 Working with activity instances

Use the SOP instance ID as a parameter to retrieve activity instances associated to this SOP via a GET request as shown in Example 5-11.

Example 5-11 Retrieving activity instances of an SOP instance

```
https://<ioc_ihs>/ibm/ioc/api/sop-service/sop/<SOP_INSTANCE_ID>/activity
```

This request returns all active activity instances associated with the SOP instance. Example 5-12 shows the response.

Example 5-12 Response sample - Retrieving activity instances

```
"uid": 598,
"required": true,
"actualStartTs": 1381321657254,
"description": "des",
"duration": 1,
"durationUnit": 3,
"durationUnitString": "durationMinutes",
"actOrder": 1,
"name": "Evacuate Building",
"statusString": "ACT_STARTED",
"status": 3,
"userOwner": "wpsadmin",
"activityDefinition": {
  "$ref": "/activity/598/activityDefinition"
},
"activityType": {
  "uid": 0,
  "description": "",
  "name": "ifThenElseActivity",
  "widget": "IfThenElseActivityTypePanel",
  "icon": "",
  "activityTypeProperty": [{
    "uid": 4,
```



```

        "name": "thenSopDefVersionUid"
    },
    {
        "uid": 5,
        "name": "elseSopDefVersionUid"
    }
  ]
},
"sop": {
  "$ref": "/sop/597",
  "name": "Launch Test Def",
  "status": 2
},
"activityProperties": [{
  "uid": 601,
  "propertyValue": "51",
  "activityTypeProperty": {
    "uid": 4,
    "name": "thenSopDefVersionUid"
  }
},
{
  "uid": 602,
  "propertyValue": "363",
  "activityTypeProperty": {
    "uid": 5,
    "name": "elseSopDefVersionUid"
  }
}],
"changeLog": {
  "$ref": "/activity/598/changeLog"
},
"comment": {
  "$ref": "/activity/598/comment"
},
"roleAccess": {
  "$ref": "/activity/598/roleAccess"
},
"reference": {
  "$ref": "/activity/598/reference"
},
"accessType": "ACCESS_READ_WRITE",
"pastDue": true,
"dueToday": false,
"dueFuture": false,
"actions": [{
  "actionName": "action.addReference",
  "actionUri": "/activity/598/reference"
},
{
  "actionName": "action.addComment",
  "actionUri": "/activity/598/comment"
}],
"actualStartTsString": "Oct 9, 2013 12:27:37 PM",
"expectedEnd": 1381245935661,
"expectedEndTsString": "Oct 8, 2013 3:25:35 PM",

```

```
"triggeredSop": null,
"canComplete": false
```

Table 5-6 shows a subset of the properties of an activity instance returned by the REST service.

Table 5-6 Subset of activity instance properties

Property	Type	Description
uid	Integer	Activity instance ID
required	Boolean	Whether this activity is required for SOP instance completion
actualStartTs	Integer	Time at which the activity started
expectedEnd	Integer	Timestamp for expected end of this activity
duration	Integer	Activity duration: number of duration units this activity takes
durationUnit	Integer	Activity duration unit. Possible values are 0 for none, 1 for minutes, 2 for hours, or 3 for days.
actOrder	Integer	Order in which this activity needs to be executed relative to other activities in the SOP. Starts at 1.
statusString	String	Activity instance status: Possible values: ACT_UNKNOWN, ACT_ACTIVE, ACT_WAITING, ACT_STARTED, ACT_SKIPPED, ACT_AUTO_STARTED, ACT_AUTO_DEFINED, ACT_AUTO_EXCEPTION, ACT_AUTO_COMPLETED, or ACT_COMPLETED
status	Integer	ACT_UNKNOWN=0 ACT_ACTIVE=1 ACT_WAITING=2 ACT_STARTED=3 ACT_SKIPPED=4 ACT_AUTO_STARTED=5 ACT_AUTO_DEFINED=6 ACT_AUTO_EXCEPTION=7 ACT_AUTO_COMPLETED=8 ACT_COMPLETED=9

The activity instance ID is required to update the activity instance using REST services.

Use the following services to change the status of an activity:

- ▶ Start an activity: `/ibm/ioc/api/sop-service/activity/<ACTIVITY_INSTANCE_ID>/start`
- ▶ Stop an activity: `/ibm/ioc/api/sop-service/activity/<ACTIVITY_INSTANCE_ID>/stop`

Note: Only automation activities can be stopped.

The following requests must be accessed via the POST method:

- ▶ Skip an activity: `/ibm/ioc/api/sop-service/activity/<ACTIVITY_INSTANCE_ID>/skip`
- ▶ Complete an activity:
`/ibm/ioc/api/sop-service/activity/<ACTIVITY_INSTANCE_ID>/complete`
- ▶ Start an If-Then-Else activity. Use one of the following requests:
 - `/ibm/ioc/api/sop-service/activity/<ACTIVITY_INSTANCE_ID>/startThen`
 - `/ibm/ioc/api/sop-service/activity/<ACTIVITY_INSTANCE_ID>/startElse`

The response to these services is an activity instance as shown in Example 5-12 on page 150 with the activity status (properties `statusString` and `status`) updated.

5.3.6 Using REST services to analyze the historical executions of an SOP

Continuous Process Improvement (CPI) requires that you analyze the historical executions of an SOP to improve the quality of the processes. Activity execution time, people involved, notifications, attached references, and comments are items that can be used to support this analysis. This data can be accessed via SOP REST APIs and consumed by a RESTful application.

One example of using the SOP REST API is to create a small application that feeds a key performance indicator (KPI) to measure the elapsed time for a specific SOP. To query the execution time of an SOP, use the REST interface to query an SOP instance as shown in Example 5-9 on page 149, parse the attribute `actualStartTs`, and calculate the time elapsed until now.

Use the same approach to measure the execution time of activities by querying activity instance properties. See Example 5-11 on page 150 for a description of how to retrieve activity instance properties.

The `actualStartTs` property can be used to learn the start time of the activity. The `expectedEnd` property can be used to check whether the activity is being executed within the planned time.

The following SOP instance properties include information that can be useful for historical analysis:

- ▶ **comment:** Example 5-13 shows a request to retrieve the comments in the SOP with an ID of `<SOP_ID>`.

Example 5-13 Retrieving comments of an SOP instance

```
https://<ioc_ahs>/ibm/ioc/api/sop-service/sop/<SOP_ID>/comment
```

- ▶ **reference:** Example 5-14 shows a request to retrieve the references in an SOP instance.

Example 5-14 Retrieving references of an SOP instance

```
https://<ioc_ibs>/ibm/ioc/api/sop-service/sop/<SOP_ID>/reference
```

Example 5-15 shows the response.

Example 5-15 Sample SOP instance references

```
{
  "uid": 367,
  "archived": false,
  "description": "NOTIFICATION",
  "name": "Fire Website",
  "priv": false,
  "uri": "http://www.firesafety.org"
},
{
  "uid": 202,
  "archived": false,
  "author": "wpsadmin",
  "description": "News Source",
  "name": "RTE News",
  "priv": false,
  "uri": "http://www.rte.ie/news"
}
```

5.4 Use case overview

During the festival in City A, several city agencies are involved in the organization and monitoring of the event. They require well-defined and coordinated processes and activities to ensure that the festival runs smoothly and city operators can react in an orderly manner if their intervention is required.

Two SOPs are implemented in this use case to illustrate how the SOP capabilities in IBM Intelligent Operations Center can be used in City A's scenario:

- ▶ **Manage event SOP:** This SOP is composed of manual activities to manage the event. Activities include positioning teams and reporting information to local authorities. City operators can constantly monitor and query the status of the activities during the execution of the SOP. This example is described in 5.5, "Manage event SOP" on page 156.
- ▶ **Manage crowd incident SOP:** This SOP is activated by a crowd incident that requires cross-agency coordination and automated tasks. This SOP includes engaging an external organization that is responsible for mobilizing security teams. The external organization has its own SOP in an external system. In this case, IBM Intelligent Operations Center integrates multiple agencies and multiple systems involved in the execution of the SOP. This example is described in 5.6, "Manage crowd incident SOP" on page 177.

5.4.1 Actors

The actors of this use case are operators of the agencies and organizations involved in the SOPs:

- ▶ **Event security:** This organization is responsible to provide the city council all relevant information regarding the event, both for planning and during the event. Relevant information includes, for example, the number of people attending the event and identified incidents.
- ▶ **City council:** The city council is responsible for managing the event and any incidents that might occur. This group is responsible for approving the SOP so that it is activated and available and can be followed during the event. The city council is also responsible for notifying other external agencies and requesting their assistance. The city council acts as the incident commander in case of incidents.
- ▶ **Emergency service:** This agency is responsible for any emergency action during the event. It must always be notified by the event organizers or the city council. It helps with health services and evacuations.

5.4.2 Desired outcome

The objectives of using IBM Intelligent Operations Center SOPs in this use case are listed:

- ▶ Manage the actions in response to events and incidents in a structured way.
- ▶ Enable multiple agencies to collaborate while responding to incidents. Agencies must be able to integrate the IBM Intelligent Operations Center SOP capabilities with their own systems to complete actions in response to events and incidents.
- ▶ Enable monitoring of SOP status.

5.4.3 IBM Intelligent Operations Center programming artifacts

This use case shows, with practical examples, how to configure and extend SOPs using the SOP capabilities in IBM Intelligent Operations Center described in 5.1, “Managing SOPs in IBM Intelligent Operations Center” on page 110, 5.2, “Configuring activities for different scenarios” on page 125, and 5.3, “SOP REST service” on page 144.

Most SOP capabilities in IBM Intelligent Operations Center are available through the SOP configuration tool in the Solution Administration portal interface. The examples also describe the use of the IBM Intelligent Operations Center extensions that can be implemented to enhance standard SOP functions.

The following artifacts are used to implement the examples in this use case:

- ▶ IBM Intelligent Operations Center SOP configuration tool used to create the SOPs
- ▶ Integration to an external system through an automation activity in an SOP
- ▶ Sample code that shows how to use the IBM Intelligent Operations Center SOP REST service to query SOP data
- ▶ An IBM Intelligent Operations Center data source with a routing expression that triggers an SOP

In this use case, the IBM Intelligent Operations Center SOP connects to the external system, IBM SmartCloud Control Desk.

5.5 Manage event SOP

The Manage event SOP includes all the required steps to prepare for and run the festival event at City A. These steps are usually identified by city officials during planning and preparation meetings where all agencies define the activities needed to operate the event.

The following activities make up this sample SOP:

1. Allocate personnel and resources.
2. Prepare the event site: cleaning, the verification of the power supply, and the deployment of equipment to operate the event.
3. Organize traffic on access roads.
4. Dispatch teams and run the event.
5. Collect information about the event and report back to the operations center, including crowd size.
6. End of the event: cleaning and recall resources.
7. Report event results.

For this hypothetical list of activities inspired by real-life scenarios, the following activity types are in the Manage event SOP:

- ▶ Manual activities: steps 1, 3, 4, 5, and 7.
- ▶ SOP activities: steps 2 and 6. These steps call child SOPs that include a list of detailed activities to prepare the event and end the event. These activities can contain agency-specific resource allocation, tasks, and actions.

The different agencies involved in the incident might want to access information about the SOP during its execution to feed their own systems. To support this requirement, this use case describes how to use the SOP REST service to query SOP data.

5.5.1 Implementation summary

Implement this use case by using the following high-level steps:

1. Create Prepare event site child SOP
2. Submit Prepare event site SOP for approval
3. Approve Prepare event site SOP
4. Create, submit for approval, and approve Close event operation child SOP
5. Create, submit for approval, and approve the parent SOP Manage event
6. Launch the Manage event SOP
7. Example of using SOP REST APIs: Querying SOP instance data

5.5.2 Implementation details

This section includes step-by-step instructions to implement this use case.

Create and approve child SOPs

As described in 5.5, “Manage event SOP” on page 156, the Manage Event SOP includes two SOP activity types which, in turn, call the following SOPs:

- ▶ Prepare event site: Includes activities to get the site ready for the event.
- ▶ Close event operation: Includes activities to clean up and recall resources after the event.

The child (subordinate) SOPs must exist before the parent SOP is configured. For more information about the SOP activity type, see 5.2.8, “Configuring SOP activities” on page 143.

Follow these steps:

1. Create the Prepare event site SOP:
 - a. Click **Solution Administration** → **Standard Operating Procedure** → **Definition**.
 - b. On the Basics tab, enter the SOP name, description, and check **Activities are done in order** if you want the activities in this SOP to be performed in sequence (see Figure 5-44). For more information about ordered activities, see 5.2.1, “Configuring activities to execute sequentially” on page 127.

Click **Next**.

The screenshot shows a web form titled "Create Standard Operating Procedure Definition". It has four tabs: "Basics", "Roles", "Activities", and "Summary". The "Basics" tab is active. Below the tabs, there is a header: "Name this procedure, and optionally add a description and references." The form is divided into several sections:

- Name:** A text input field containing "Prepare event site".
- Description:** A larger text area containing "Steps to prepare the event location prior to the start of the event.".
- General Settings:** A section with a checked checkbox labeled "Activities are done in order".
- References:** A section with a table for adding references. The table has two columns: "Name" and "Description". Below the table is an "Add Reference" button.

At the bottom of the form, there are three buttons: "Back", "Next", and "Cancel".

Figure 5-44 Prepare event site SOP definition - Basic information

- c. Add the roles responsible for monitoring the execution (Readers) and completion (Owners) of the Prepare event site SOP (see Figure 5-45):
 - i. Select **CityWideAdmin** in the Available Roles list and click **Add**. The CityWideAdmin group moves to Selected Roles. Select **Owners**.
 - ii. Select **CityWideOperator** in the Available Roles list and click **Add**. The CityWideAdmin group moves to Selected Roles. Select **Owners**.

Note: Only users that belong to a group with the Owners role in the SOP definition have authority to approve the SOP. For more information about roles and actions, see Table 5-1 on page 114.

- iii. Click **Next**.

Create Standard Operating Procedure Definition

Basics **Roles** Activities Summary

Select the roles that are responsible for monitoring the execution and completion of this procedure.

Available Roles:

- CityWideExecutive
- CityWideSupervisor
- stadmins
- wcmadmins
- wpsadmins
- wpsContentAdministrators
- wpsDocReviewer

Add

Selected Roles:

Role	Owners	Readers
CityWideAdmin	<input checked="" type="radio"/>	<input type="radio"/>
CityWideOperator	<input checked="" type="radio"/>	<input type="radio"/>

Remove

Figure 5-45 Prepare event site SOP definition - Roles

- d. Add activities to the Prepare event site SOP. Add one manual activity for each of the following steps:
 - Clean public areas
 - Verify power supply
 - Deploy operational equipment
 Repeat the next steps for each activity:
 - i. On the Activities tab, click **Add**.

- ii. Complete the name, duration, and activity type fields in the Create Activity Definition window as shown in Figure 5-46.
- iii. Click **Add** to add the next activity.

Create Activity Definition

Required

Name * Clean public areas

Roles
Owners: CityWideOperator,
Readers: CityWideAdmin,

Duration | None

Description

Activity type Manual Activity

Add Cancel

Figure 5-46 Prepare event site SOP definition - Adding manual activities

- e. After you finish adding all the activities to the SOP, review the list of activities and click **Next** (see Figure 5-47 on page 160).

Create Standard Operating Procedure Definition

Basics Roles **Activities** Summary

Create a list of the activities that this standard operating procedure contains.

Add Edit Delete

- ▶ 1: Clean public areas
- ▶ 2: Verify power supply
- ▶ 3: Deploy equipments

Back **Next** **Cancel**

Figure 5-47 Prepare event site SOP definition - Activities list

- f. On the Summary page, click **Save** to save the Prepare event site SOP as a draft.
2. Submit the Prepare event site SOP draft for approval.

The Prepare event site SOP definition is listed in the Standard Operating Procedures: Definition window. Select the **Prepare event site** and click **Submit for Approval** (see Figure 5-48 on page 161). For more information, see 5.1.4, “Working with drafts and submitting for approval” on page 117.

The SOP draft must now go through a formal approval process.

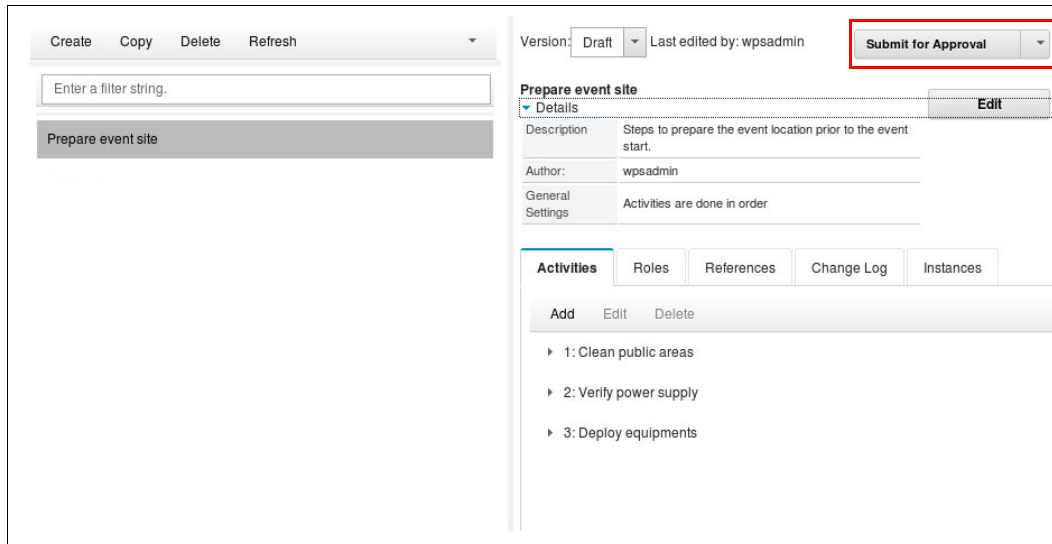


Figure 5-48 Submitting Prepare event site SOP draft for approval

3. Approve the Prepare event site SOP draft.

This task must be performed by users with the Owners role in the SOP definition. For more information, see 5.1.5, “Approving or disapproving drafts” on page 118.

The Prepare event site SOP definition is listed in the Standard Operating Procedures: Definition window. Select **Prepare event site** and click **Approve** (see Figure 5-49).

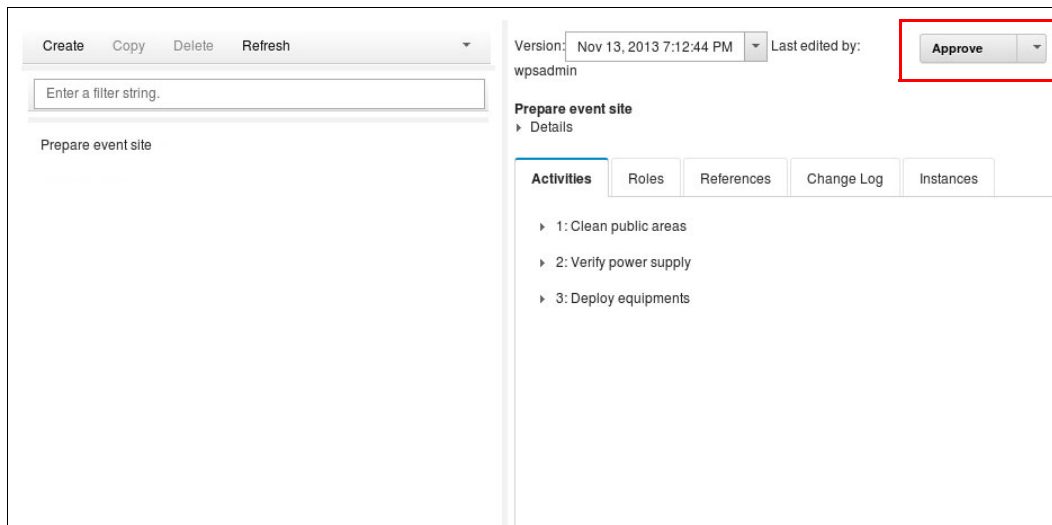


Figure 5-49 Prepare event site SOP is ready for approval

The Prepare event site SOP is now activated and can be called by the parent SOP (see Figure 5-50 on page 162).

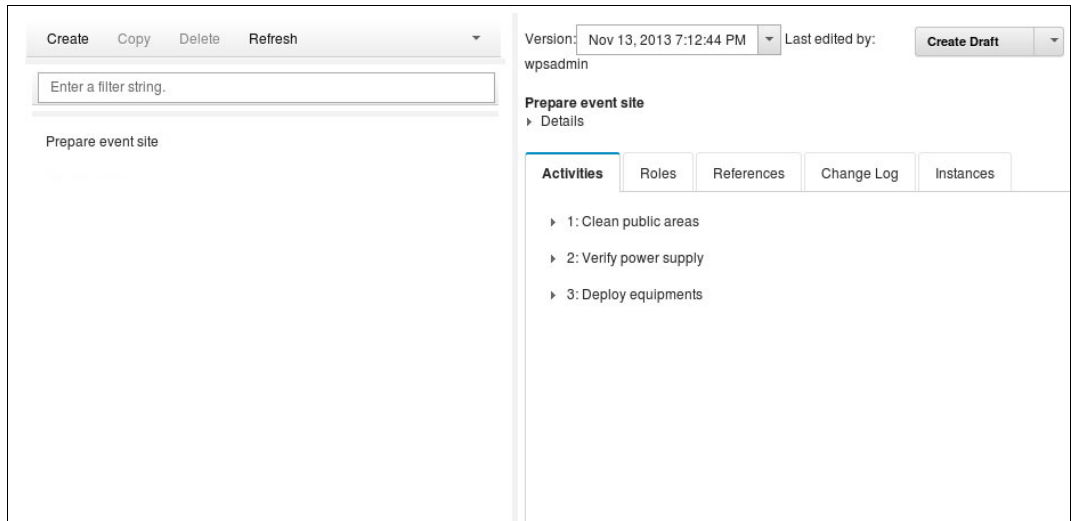


Figure 5-50 Prepare event site SOP

4. Create, submit for approval, and approve the Close event operation SOP.

Follow steps 1 on page 157 through step 3 on page 161 for the Close event operation SOP.

After you complete the steps, the Close event operation SOP is activated and can be called by the parent SOP (see Figure 5-51).

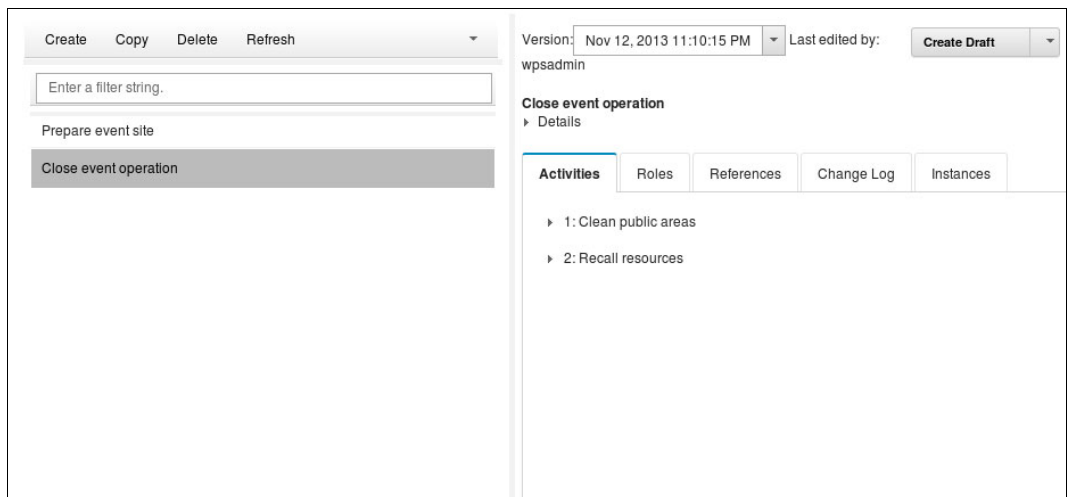


Figure 5-51 Close event operation SOP

Create and approve the parent SOP

This section describes the steps to create the Manage event parent SOP:

1. Create the Manage event site SOP:
 - a. Click **Solution Administration** → **Standard Operating Procedure** → **Definition**.
 - b. On the Basics tab, enter the SOP name, description, and check **Activities are done in order** if you want the activities in this SOP to be performed in sequence (see Figure 5-52 on page 163).
Click **Next**.

Create Standard Operating Procedure Definition

Basics Roles Activities Summary

Name this procedure, and optionally add a description and references.

Name *	Manage event						
Description	The Manage Event SOP covers all steps required to prepare for and run the event at City A.						
General Settings	<input checked="" type="checkbox"/> Activities are done in order						
References	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%; border: 1px solid #ccc;"></th> <th style="width: 30%; border: 1px solid #ccc;">Name</th> <th style="width: 40%; border: 1px solid #ccc;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="text-align: center; padding: 5px;">Add Reference</td> </tr> </tbody> </table>		Name	Description	Add Reference		
	Name	Description					
Add Reference							

Back
Next
Cancel

Figure 5-52 Manage event SOP definition - Basic information

- c. Add the roles responsible for monitoring the execution (Readers) and completion (Owners) of the Manage event site SOP (see Figure 5-53 on page 164):
 - i. Select **CityWideAdmin** in the Available Roles list and click **Add**. The CityWideAdmin group moves to Selected Roles. Select **Owners**.
 - ii. Select **CityWideOperator** in the Available Roles list and click **Add**. The CityWideAdmin group moves to Selected Roles. Select **Owners**.

Note: Only users that belong to a group with the Owners role have authority to approve the SOP. For more information about roles and actions, see Table 5-1 on page 114.

- iii. Click **Next**.

Create Standard Operating Procedure Definition

Basics **Roles** Activities Summary

Select the roles that are responsible for monitoring the execution and completion of this procedure.

Available Roles:

Enter a filter string.

- CityWideExecutive
- CityWideSupervisor
- stadmins
- wcadmins
- wpsadmins
- wpsContentAdministrators
- wpsDocReviewer

Add

Selected Roles:

Role	Owners	Readers
CityWideAdmin	<input checked="" type="radio"/>	<input type="radio"/>
CityWideOperator	<input checked="" type="radio"/>	<input type="radio"/>

Remove

Back

Next

Cancel

Figure 5-53 Manage event SOP definition - Roles

- d. Add a manual activity to the Manage event site SOP.

Add the **Allocate personnel and resources** manual activity described in 5.5, “Manage event SOP” on page 156 (see Figure 5-54 on page 165).

Create Activity Definition

Required	<input checked="" type="checkbox"/>
Name *	Allocate personnel and resources
Roles	Owners: CityWideOperator Readers: CityWideAdmin
Duration	<input type="text"/> None
Description	<input type="text"/>
Activity type	Manual Activity

Figure 5-54 Manage event SOP definition - Adding Manual Activity

- e. Add an SOP activity to launch the Prepare event site SOP. Select **SOP Activity** in the Activity type pull-down list (see Figure 5-55 on page 166).

Create Activity Definition

Required	<input type="checkbox"/>				
Name *	Prepare for event				
Roles	Owners: CityWideOperator Readers: CityWideAdmin				
Duration	2 Days				
Description	Execute SOP for event preparation				
Activity type	SOP Activity				
	Standard operating procedure: Prepare event site				
References	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">Add Reference</td> </tr> </tbody> </table>	Name	Description	Add Reference	
Name	Description				
Add Reference					

Figure 5-55 Manage event SOP - Adding SOP activity to launch the Prepare event site SOP

- f. Add one Manual activity (Activity type Manual) for each of the following tasks:
 - Organize traffic
 - Dispatch teams
 - Collect information
- g. Add an SOP activity to launch the Close event operation SOP. Select **SOP Activity** in the Activity type pull-down list (see Figure 5-56 on page 167).

Create Activity Definition

Required	<input type="checkbox"/>				
Name *	Close event operation				
Roles	Owners: CityWideOperator Readers: CityWideAdmin				
Duration	2 Days				
Description					
Activity type	SOP Activity				
	Standard operating procedure: Close event operation				
References	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">Add Reference</td> </tr> </tbody> </table>	Name	Description	Add Reference	
Name	Description				
Add Reference					

Figure 5-56 Manage event SOP - Adding SOP activity to launch the Close event operation SOP

- h. Add the Report event results manual activity (described in 5.5, “Manage event SOP” on page 156).
- i. Review the list of activities and click **Next** (see Figure 5-57 on page 168).

Create Standard Operating Procedure Definition

Basics Roles **Activities** Summary

Create a list of the activities that this standard operating procedure contains.

Add Edit Delete



- ▶ 1: Allocate personnel and resources
- ▶ 2: Prepare for event 
- ▶ 3: Organize traffic on access roads
- ▶ 4: Dispatch teams
- ▶ 5: Collect event information
- ▶ 6: Close event operation 
- ▶ 7: Report event results

Figure 5-57 Manage event SOP definition - Activity list

- j. On the Summary page, click **Save** to save the Manage event SOP as a draft.
2. Submit the Manage event SOP draft for approval.

The Manage event SOP definition is listed in the Standard Operating Procedures: Definition window. Select **Manage event** and click **Submit for Approval** (see Figure 5-58 on page 169). For more information, see 5.1.4, “Working with drafts and submitting for approval” on page 117.

The SOP draft must now go through a formal approval process.

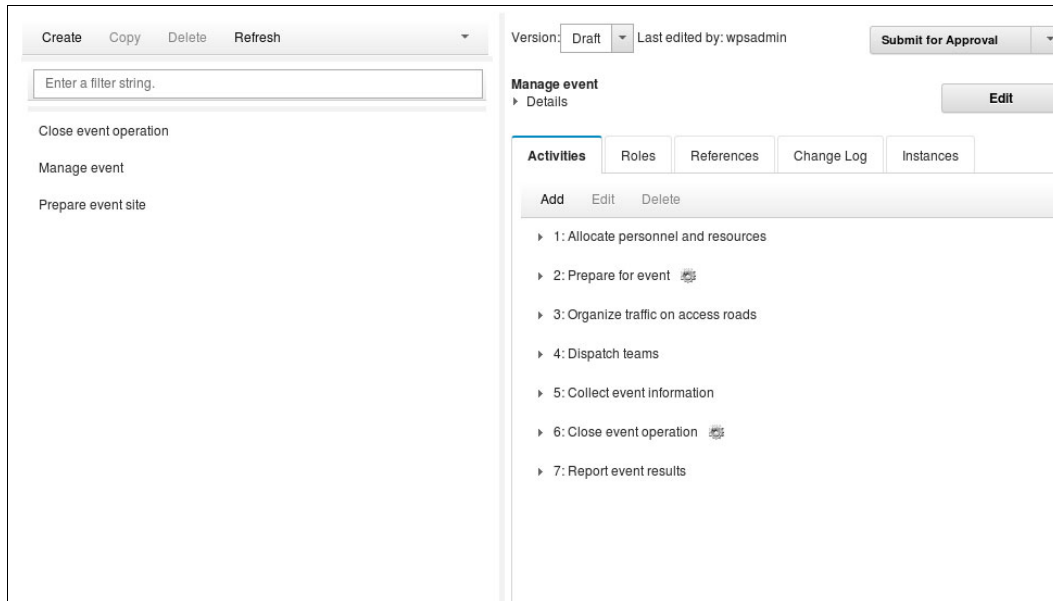


Figure 5-58 Submitting Manage event SOP draft for approval

3. Approve the Manage event SOP draft.

This task must be performed by users with the Owners role in the SOP definition. For more information, see 5.1.5, “Approving or disapproving drafts” on page 118.

The Manage event SOP definition is listed in the Standard Operating Procedures: Definition window. Select **Manage event** and click **Approve** (see Figure 5-59).

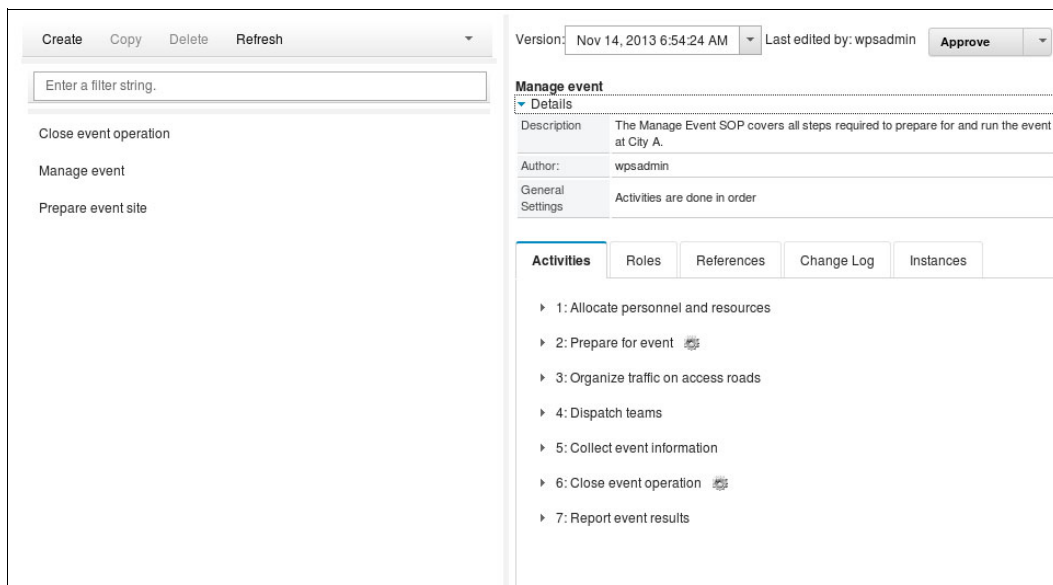


Figure 5-59 Approving Manage event SOP

The Manage event SOP is now activated and can be launched (see Figure 5-60 on page 170).

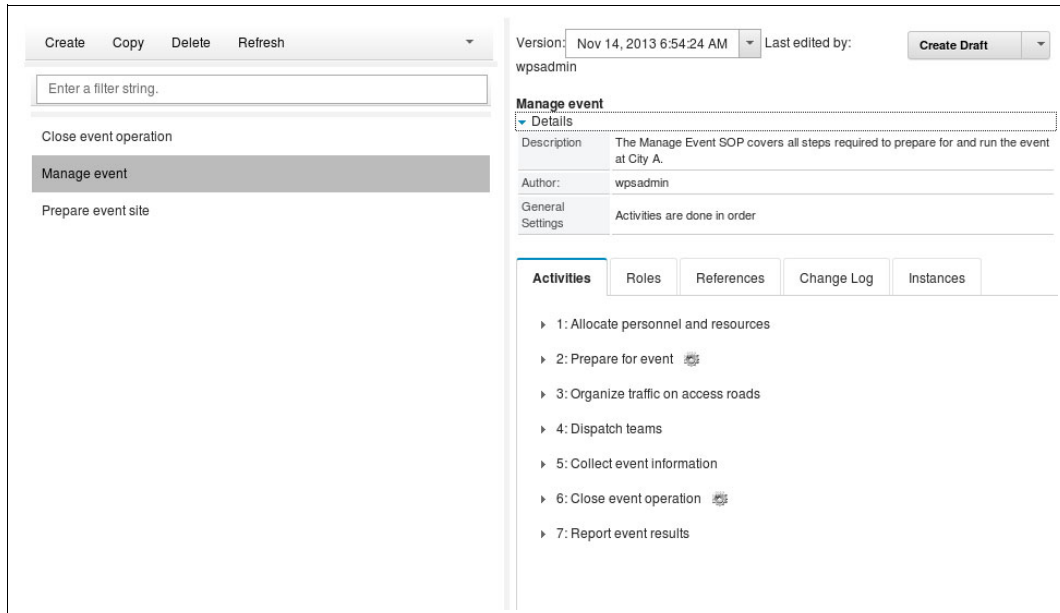


Figure 5-60 Manage event SOP

Launch the SOP

Defining the SOP and its activities occurs a long time before the event. At some point, when the activities that make up the SOP must be executed, an instance of the SOP is launched.

To launch the SOP, users must have the required role as described in Table 5-1 on page 114. For more information about launching SOPs, see 5.1.6, “Launching SOPs” on page 119.

To launch the SOP, from the Solution Administration view, expand **Standard Operating Procedures** → **Definition** → **Manage event** → **Launch** from the Definition Actions pull-down list (see Figure 5-61).

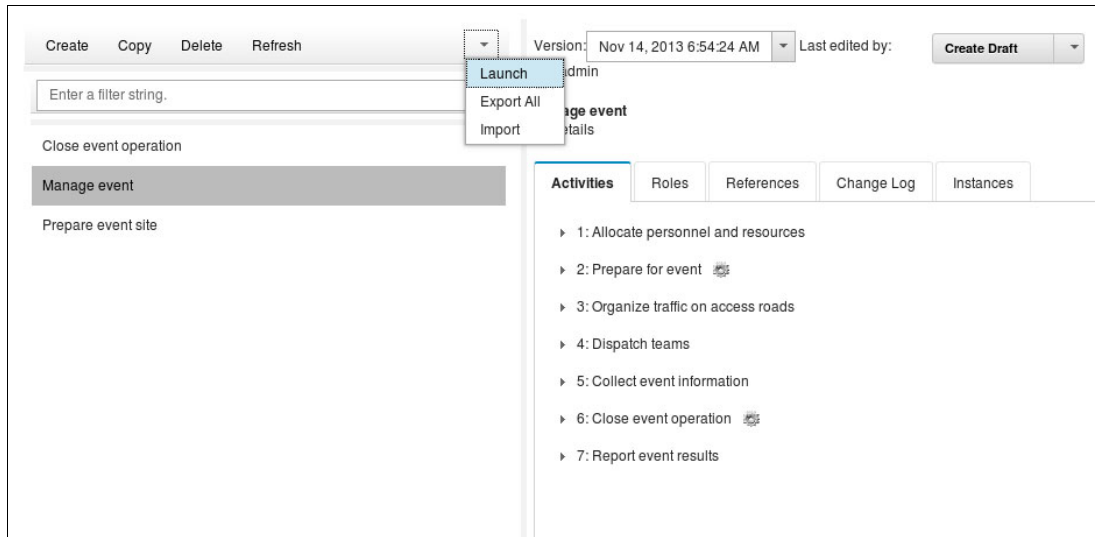


Figure 5-61 Launching the Manage event SOP

The SOP is started and an instance of it is created. Users with Owners and Readers roles can access the SOP instance through the My Activities menu.

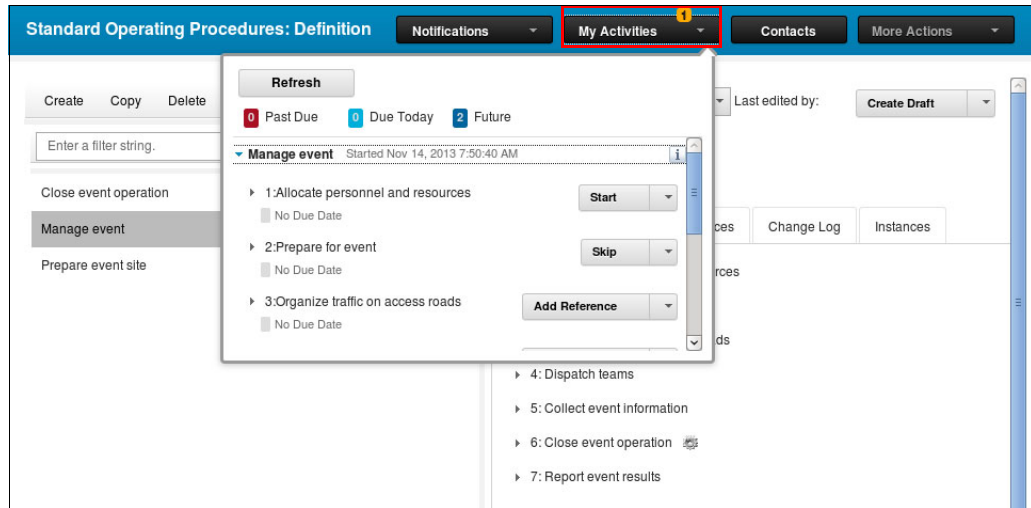


Figure 5-62 My Activities menu with a Manage event SOP instance

Users with the Owners role can start and complete the manual activities in the SOP.

To start the first activity, click **Start** next to 1: Allocate personnel and resources. As in the Manage event SOP, the activities are performed in order. The next activity is ready to start only if the previous activity is marked Complete. For required activities, the Start button is enabled after the previous activity is completed. For non-required activities, a Skip button is available to skip the particular activity if the operator decides not to run it (see Figure 5-63).

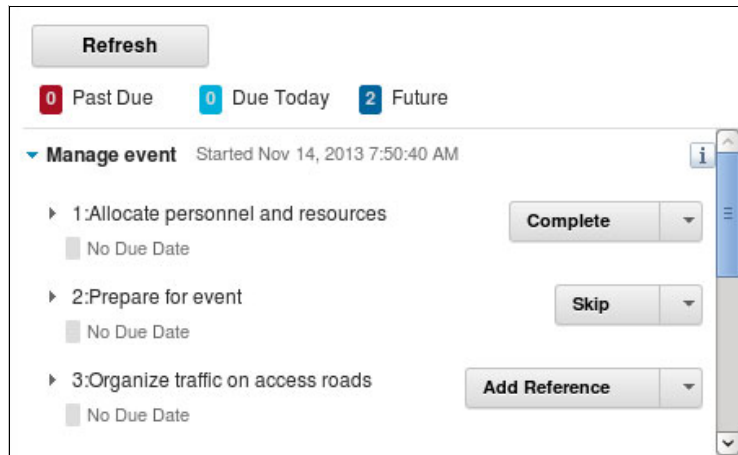


Figure 5-63 Starting, completing, and skipping activities

5.5.3 Querying SOP instance data

Now that there is an active instance of the Manage event SOP, it is possible to query data about the running SOP instance using the SOP REST service. This capability is useful in cases where the client requires information about the running SOPs that is not provided by the IBM Intelligent Operations Center UI. Also, external systems can consume data from running SOPs by querying the SOP instance via the SOP REST interface.

For example, the following steps describe how to get the status of one particular activity in the Manage event SOP, the Allocate personnel and resources activity:

1. Query the SOP definitions available in IBM Intelligent Operations Center with a GET method as shown in Example 5-16. All SOP definitions in IBM Intelligent Operations Center are returned in the response, independent of their status (for example, approved, in draft, and so on).

Example 5-16 Accessing the list of SOP definitions

SOP definition endpoint:

`https://<ioc_ihs>/ibm/ioc/api/sop-service/sopDefinition/`

Response:

```
[{
  "matchAll": false,
  "trigAutomatic": false,
  "uid": 106,
  "cancelEvent": false,
  "archived": false,
  "author": "wpsadmin",
  "description": "The Manage Event SOP covers all steps required to prepare
for and run the event at City A.",
  "name": "Manage event",
  "ordered": true,
  "statusString": "SOP_APPROVED",
  "status": 3,
  "transitional": false,
  "version": 1384430064761,
  "versionUid": 105,
  "sop": {
    "$ref": "/sopDefinition/106/sop"
  },
  "versions": {
    "$ref": "/sopDefinition/106/versions"
  },
  "instance": {
    "$ref": "/sopDefinition/106/instance"
  },
  "changeLog": {
    "$ref": "/sopDefinition/106/changeLog"
  },
  "triggerParameter": {
    "$ref": "/sopDefinition/106/triggerParameter"
  },
  "reference": {
    "$ref": "/sopDefinition/106/reference"
  },
  "roleAccess": {
    "$ref": "/sopDefinition/106/roleAccess"
  },
  "activityDefinition": {
    "$ref": "/sopDefinition/106/activityDefinition"
  },
  "versionString": "Nov 14, 2013 11:54:24 AM",
  "activeInstanceCount": 0,
}
```

```

    "accessType": "ACCESS_READ_WRITE",
    "actions": [{
      "actionName": "action.addReference",
      "actionUri": "sopDefinition/106/reference"
    },
    {
      "actionName": "action.createDraft",
      "actionUri": "sopDefinition/106/createDraft"
    }
  ]
},
{
  ...
}]

```

2. Parse the JSON response, access the **name** property with value “Manage event” and get the corresponding SOP definition ID, the **uid** property.

In this example, **name**=Manage event and corresponding **uid**=106. This uid is used to query the active instances of the Manage event SOP.

3. Query the active instances in the SOP definition with **uid**=106. Use a GET method as shown in Example 5-17.

There is one active instance running, identified by the **uid**=146. This instance was started when the SOP was launched as described in “Launch the SOP” on page 170.

Example 5-17 Accessing Manage event SOP running instances

SOP instance endpoint:

https://<ioc_ihs>/ibm/ioc/api/sop-service/sopDefinition/106/instance

Response:

```

[ {
  "uid": 146,
  "cancel_Event": false,
  "archived": false,
  "description": "The Manage Event SOP covers all steps required to prepare for
and run the event at City A.",
  "name": "Manage event",
  "ordered": true,
  "statusString": "SOP_STARTED",
  "status": 2,
  "activity": {
    "$ref": "/sop/146/activity"
  },
  "sopDefinition": {
    "$ref": "/sopdefinition/106"
  },
  "changeLog": {
    "$ref": "/sop/146/changeLog"
  },
  "comment": {
    "$ref": "/sop/146/comment"
  },
  "reference": {
    "$ref": "/sop/146/reference"
  },
} ],

```

```

"roleAccess": {
  "$ref": "/sop/146/roleAccess"
},
"creator": "wpsadmin",
"actualStartTs": 1384433440013,
"accessType": "ACCESS_READ_WRITE",
"actions": [{
  "actionName": "action.stop",
  "actionUri": "/sop/146/stop"
},
{
  "actionName": "action.addReference",
  "actionUri": "/sop/146/reference"
},
{
  "actionName": "action.exportSopInstance",
  "actionUri": "exportsopinstance/146"
},
{
  "actionName": "action.addComment",
  "actionUri": "/sop/146/comment"
}],
"event": {
  "$ref": "/sop/146/event"
},
"actualStartTsString": "Nov 14, 2013 12:50:40 PM"
}]

```

4. Use the GET method to query the list of activities in the SOP instance with `uid=146` as shown in Example 5-18 on page 175.

Parse the JSON response to access the properties of the activities including the `uid`, current status of the activity, and the time when it was started.

The activity Allocate personnel and resources, with `uid = 157`, was started, as shown in Figure 5-62 on page 171. The status information can be accessed from the `statusString` and `status` properties, with the values `ACT_STARTED` and `3`, respectively.

Example 5-18 Accessing activities of the SOP instance

Activities instances endpoint:

https://<ioc_ihs>/ibm/ioc/api/sop-service/sop/146/activity

Response

```
[{
  "uid": 157,
  "required": true,
  "actualStartTs": 1384433707507,
  "description": "",
  "duration": 0.0,
  "durationUnit": 0,
  "durationUnitString": "durationNone",
  "actOrder": 1,
  "name": "Allocate personnel and resources",
  "statusString": "ACT_STARTED",
  "status": 3,
  "userOwner": "wpsadmin",
  "activityDefinition": {
    "$ref": "/activity/157/activityDefinition"
  },
  "sop": {
    "$ref": "/sop/146",
    "name": "Manage event",
    "status": 2
  },
  "activityProperties": [],
  "changeLog": {
    "$ref": "/activity/157/changeLog"
  },
  "comment": {
    "$ref": "/activity/157/comment"
  },
  "roleAccess": {
    "$ref": "/activity/157/roleAccess"
  },
  "reference": {
    "$ref": "/activity/157/reference"
  },
  "accessType": "ACCESS_READ_WRITE",
  "pastDue": false,
  "dueToday": false,
  "dueFuture": false,
  "actions": [{
    "actionName": "action.complete",
    "actionUri": "/activity/157/complete"
  },
  {
    "actionName": "action.addReference",
    "actionUri": "/activity/157/reference"
  },
  {
    "actionName": "action.addComment",
    "actionUri": "/activity/157/comment"
  }
]
```

```

    ]],
    "actualStartTsString": "Nov 14, 2013 12:55:07 PM",
    "expectedEnd": 0,
    "expectedEndTsString": "Jan 1, 1970 12:00:00 AM",
    "triggeredSop": null,
    "canComplete": false
  },
  {
    "uid": 147,
    "required": false,
    "description": "Execute SOP for event preparation",
    "duration": 2.0,
    "durationUnit": 1,
    "durationUnitString": "durationDays",
    "actOrder": 2,
    "name": "Prepare for event",
    "statusString": "ACT_ACTIVE",
    "status": 1,
    "activityDefinition": {
      "$ref": "/activity/147/activityDefinition"
    },
    "activityType": {
      "uid": 3,
      "description": "",
      "name": "sopActivity",
      "widget": "SopActivityTypePanel",
      "icon": ""
    },
    "sop": {
      "$ref": "/sop/146",
      "name": "Manage event",
      "status": 2
    },
    "activityProperties": [{
      "uid": 149,
      "propertyValue": "26",
      "activityTypeProperty": {
        "uid": 3,
        "name": "sopDefVersionUid"
      }
    }
  ]],
    "changeLog": {
      "$ref": "/activity/147/changeLog"
    },
    "comment": {
      "$ref": "/activity/147/comment"
    },
    "roleAccess": {
      "$ref": "/activity/147/roleAccess"
    },
    "reference": {
      "$ref": "/activity/147/reference"
    },
    "accessType": "ACCESS_READ_WRITE",
    "pastDue": false,

```

```

    "dueToday": false,
    "dueFuture": true,
    "actions": [{
      "actionName": "action.skip",
      "actionUri": "/activity/147/skip"
    },
    {
      "actionName": "action.addReference",
      "actionUri": "/activity/147/reference"
    },
    {
      "actionName": "action.addComment",
      "actionUri": "/activity/147/comment"
    }
  ]],
  "expectedEnd": 0,
  "triggeredSop": null,
  "canComplete": false
},
{
  ...
}]

```

5.6 Manage crowd incident SOP

The Manage crowd incident SOP is triggered when a crowd incident is detected. This SOP includes the steps to respond to the incident during the festival at City A. These steps are usually identified by city officials, local law enforcement, or other agency representatives responsible for responding to a crowd incident. The city agencies define the list of activities needed based on predefined procedures, preferred practices, and their own previous experience when responding to this type of incident.

The following list shows the activities that make up the Manage crowd incident SOP:

1. Collect initial information about the incident: location, estimated crowd size, and number of affected people.
2. Prepare security personnel.
3. Dispatch security teams to provide support locally.
4. In case of accidents and injuries, request medical emergency services support:
 - a. Dispatch medical teams.
 - b. Provide medical assistance.
5. Organize evacuation routes.

For this hypothetical list of activities inspired by real-life scenarios, the following activity types are used:

- ▶ Manual activities: steps 1, 3, and 5.
- ▶ Automation activity: step 2. This activity invokes another procedure on an external system. In this example, IBM SmartCloud Control Desk is used as the external system and the Manage crowd incident SOP in IBM Intelligent Operations Center starts a job plan in IBM SmartCloud Control Desk that includes the details for providing medical assistance.

Note: This use case requires IBM SmartCloud Control Desk installed and configured to run a job plan. It must also be configured for integration with IBM Intelligent Operations Center.

IBM SmartCloud Control Desk 7.5.1 is provided in the box with IBM Intelligent Operations Center 1.6 Extended Edition, but it is not installed and configured by the installation scripts.

- ▶ If-then-else activity: step 4. This activity calls another SOP (Provide medical support) for providing medical emergency support in the case of injuries. The child SOP consists of manual activities to perform steps 4.a and 4.b.

5.6.1 Implementation summary

Use the following high-level tasks to implement this use case:

1. Create a job plan in IBM SmartCloud Control Desk to manage assistance in medical emergencies.
2. Configure the integration between IBM SmartCloud Control Desk and IBM Intelligent Operations Center. For details, see 9.2, “Integration with IBM SmartCloud Control Desk through SOP automation activity” on page 322.
3. Create and approve child SOP Provide medical support.
4. Create and approve parent SOP Manage crowd incident:
 - a. Add manual activities.
 - b. Add SOP activity to call Provide medical support child SOP.
 - c. Add automation activity to invoke the sample job plan Prepare personnel for standby (prepare) in IBM SmartCloud Control Desk.
5. Launch the Manage crowd incident SOP with a routing expression in a data source.

5.6.2 Implementation details

This section includes step-by-step instructions to implement this use case.

Create a job plan in IBM SmartCloud Control Desk

Note: This use case requires a server with IBM SmartCloud Control Desk installed and configured. For information about IBM SmartCloud Control Desk installation and configuration, see the IBM SmartCloud Control Desk Version 7.5.1 Information Center:

<http://bit.ly/1ko0FDB>

From IBM SmartCloud Control Desk Start Center, click **Go To** → **Planning** → **Job Plans**. Create a job plan that will be initiated and tracked by an SOP in IBM Intelligent Operations Center.

Note: IBM SmartCloud Control Desk is not part of the standard installation of IBM Intelligent Operations Center and it is not required for basic SOP functions.

In IBM Intelligent Operations Center 1.6, you can create SOPs with the built-in SOP configuration tool, except for complex automated workflow SOPs. If you require this type of SOP, you need to install and configure IBM SmartCloud Control Desk, which is provided in the box with IBM Intelligent Operations Center 1.6.

If you are already using IBM Tivoli Service Request Manager or IBM SmartCloud Control Desk in your environment, you can follow the steps describe in this use case to integrate the existing job plans with IBM Intelligent Operations Center via SOPs.

Configure the IBM SmartCloud Control Desk integration

For information about configuring IBM Intelligent Operations Center for connection to IBM SmartCloud Control Desk, see 9.2, “Integration with IBM SmartCloud Control Desk through SOP automation activity” on page 322.

Tip: When creating the job plan in IBM SmartCloud Control Desk, match the criteria defined in the `tcrm.listAutomation.whereClause` system property in IBM Intelligent Operations Center. As described in 9.2, “Integration with IBM SmartCloud Control Desk through SOP automation activity” on page 322, the example criteria used in this book starts the job plan number with the string PLUS1.

Create and approve child SOP

As described in 5.6, “Manage crowd incident SOP” on page 177, the Manage crowd incident SOP includes one SOP activity type, which calls the Provide medical support SOP. The child (subordinate) SOP must exist before the parent SOP is configured. For more information about the SOP activity type, see 5.2.8, “Configuring SOP activities” on page 143.

Follow these steps:

1. Create the Provide medical support SOP:
 - a. Click **Solution Administration** → **Standard Operating Procedures** → **Definition**.
 - b. On the Basics tab, enter the SOP name, description, and check **Activities are done in order** if you want the activities in this SOP to be performed in sequence (see Figure 5-64 on page 180). For more information about ordered activities, see 5.2.1, “Configuring activities to execute sequentially” on page 127.

Click **Next**.

Create Standard Operating Procedure Definition

Basics
Roles
Activities
Summary

Name this procedure, and optionally add a description and references.

Name *

Description

General Settings

References

Provide medical support

This SOP includes the tasks assigned to emergency services for providing medical emergencies assistance in case of accidents and injuries.

Activities are done in order

	Name	Description
Add Reference		

Back
Next
Cancel

Figure 5-64 Provide medical support SOP definition - Basic information

- c. Add the roles responsible for monitoring the execution (Readers) and completion (Owners) of the Provide medical support SOP (see Figure 5-65 on page 181):
 - i. Select **CityWideAdmin** in the Available Roles list and click **Add**. The CityWideAdmin group moves to Selected Roles. Select **Owners**.
 - ii. Select **CityWideOperator** in the Available Roles list and click **Add**. The CityWideOperator group moves to Selected Roles. Select **Owners**.

Note: Only users that belong to a group with the Owners role have authority to approve the SOP. For more information about roles and actions, see Table 5-1 on page 114.

- iii. Click **Next**.

Create Standard Operating Procedure Definition

Basics **Roles** Activities Summary

Select the roles that are responsible for monitoring the execution and completion of this procedure.

Available Roles:

Enter a filter string.
CityWideExecutive
CityWideSupervisor
stadmins
wcmadmins
wpsadmins
wpsContentAdministrators
wpsDocReviewer

Add

Selected Roles:

Role	Owners	Readers
CityWideAdmin	<input checked="" type="radio"/>	<input type="radio"/>
CityWideOperator	<input checked="" type="radio"/>	<input type="radio"/>

Remove

Back
Next
Cancel

Figure 5-65 Provide medical support SOP definition - Roles

d. Add activities to the Provide medical support SOP. Add one manual activity for each of the following steps:

- Dispatch medical teams
- Provide medical assistance

Repeat the next steps for each activity:

- i. On the Activities tab, click **Add**.
- ii. Complete the name, roles, and activity type in the Activity Definition windows as shown in Figure 5-66 on page 182.
- iii. Click **Add** to add the next activity.

Create Activity Definition

Required	<input checked="" type="checkbox"/>
Name *	Dispatch medical teams
Roles	Owners: CityWideOperator Readers: CityWideAdmin
Duration	None
Description	
Activity type	Manual Activity

Figure 5-66 Provide medical support SOP definition - Adding manual activities

- e. After you finish adding all the activities in the SOP definition, review the list of activities and click **Next** (see Figure 5-67).

Create Standard Operating Procedure Definition

Basics Roles **Activities** Summary

Create a list of the activities that this standard operating procedure contains.

 Edit Delete

- ▶ 1: Dispatch medical teams
- ▶ 2: Give medical help

Figure 5-67 Provide medical support SOP definition - Activities list

2. Submit the Provide medical support SOP draft for approval.

The Provide medical support SOP definition is listed in the Standard Operating Procedures: Definition window. Select **Provide medical support** and click **Submit for Approval** (see Figure 5-68 on page 183). For more information, see 5.1.4, “Working with drafts and submitting for approval” on page 117.

The SOP draft must now go through a formal approval process.

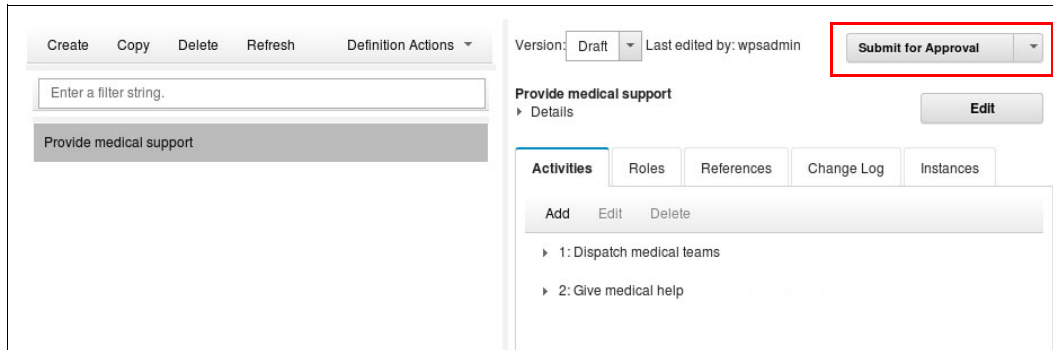


Figure 5-68 Submitting Provide medical support SOP draft for approval

3. Approve the Provide medical support SOP draft.

This task must be performed by users with the Owners role in the SOP definition. For more information, see 5.1.5, “Approving or disapproving drafts” on page 118.

The Provide medical support SOP definition is listed in the Standard Operating Procedures: Definition window. Select **Provide medical support** and click **Approve** (see Figure 5-69).

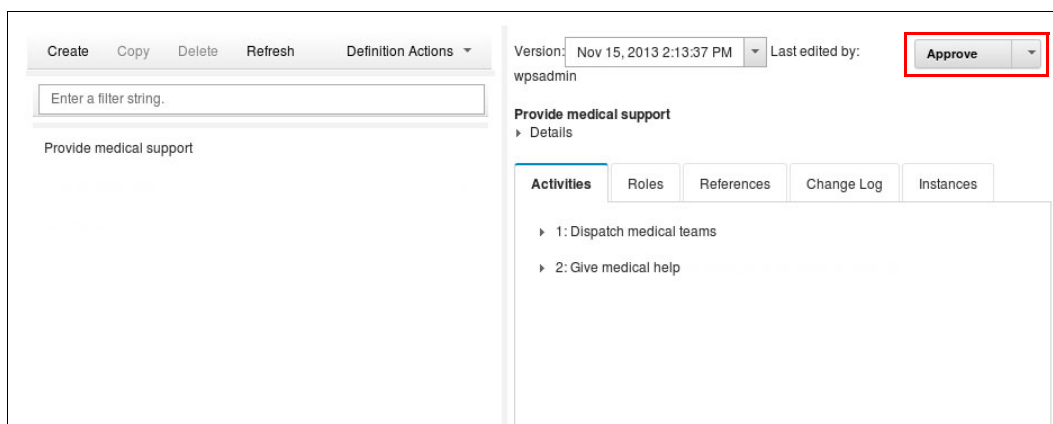


Figure 5-69 Provide medical support SOP ready for approval

The Provide medical support SOP is now activated and can be called by the parent SOP.

Create and approve the parent SOP

This section describes the steps to create the Manage crowd incident SOP. Follow these steps:

1. Create the Manage crowd incident SOP:
 - a. Click **Solution Administration** → **Standard Operating Procedure** → **Definition**.
 - b. On the Basics tab, enter the SOP name, description, and check **Activities are done in order** if you want the activities in this SOP to be performed in sequence (see Figure 5-70 on page 184).
Click **Next**.

Create Standard Operating Procedure Definition

Basics
Roles
Activities
Summary

Name this procedure, and optionally add a description and references.

Name *

Description

General Settings

References

Manage crowd incident

This SOP covers the response actions required to respond to a crowd incident.

Activities are done in order

	Name	Description
Add Reference		

Back
Next
Cancel

Figure 5-70 Manage crowd incident SOP definition - Basic information

- c. Add the Manual activities 1, 3, and 5 described in 5.6, “Manage crowd incident SOP” on page 177. Figure 5-71 on page 185 shows the Collect initial information about the incident manual activity.

Create Activity Definition

Required	<input checked="" type="checkbox"/>				
Name *	Collect initial information about the incident				
Roles	Owners: CityWideOperator Readers: CityWideAdmin				
Duration	None				
Description	Gather data location, impact, crowd size, injured people.				
Activity type	Manual Activity				
References	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">Add Reference</td> </tr> </tbody> </table>	Name	Description	Add Reference	
Name	Description				
Add Reference					

Figure 5-71 Manage crowd incident SOP definition - Adding Manual activity

- d. Create the Automation activity to invoke the IBM SmartCloud Control Desk job plan created in “Create a job plan in IBM SmartCloud Control Desk” on page 178 (see Figure 5-72 on page 186):
 - i. Select **Automation Activity** in the Activity type pull-down list.
 - ii. Select the **Prepare personnel for standby (prepare)** job plan (example for this use case) that this activity will invoke in IBM SmartCloud Control Desk.

Create Activity Definition

Required	<input checked="" type="checkbox"/>
Name *	Prepare security personnel
Roles	Owners: CityWideOperator Readers: CityWideAdmin
Duration	None
Description	
Activity type	Automation Activity
Job plan:	Evacuate impacted areas (Respond and Recover) Evacuate impacted areas (Respond and Recover) Prepare for severe weather evacuation (Prepare) Prepare personnel for standby (Prepare) Initial preparation for severe weather (Mitigate) Provide medical help
References	<input type="text"/> <input type="text"/>

Figure 5-72 Automation activity and the IBM SmartCloud Control Desk job plan to invoke

- e. Create an If-Then-Else activity. After a manual assessment of the incident, if medical assistance is required, the operator launches the Provide medical support SOP:
 - i. Select **If-Then-Else Activity** in the Activity type pull-down list.
 - ii. For the Then: clause, select the **Provide medical support** job plan (example for this use case) that this activity will invoke in IBM SmartCloud Control Desk (see Figure 5-73 on page 187).

Create Activity Definition

Required	<input checked="" type="checkbox"/>
Name *	Assess if there are people injured
Roles	Owners: CityWideOperator Readers: CityWideAdmin
Duration	None
Description	
Activity type	If-Then-Else Activity Then: Provide medical support Else: [None]

Figure 5-73 If-Then-Else activity to invoke Provide medical support job plan if required

- f. Review the list of activities and click **Next**.
 - g. On the Summary page, click **Save** to save the Manage crowd incident SOP as a draft.
2. Submit the Manage crowd incident SOP draft for approval.

The Manage crowd incident SOP definition is listed in the Standard Operating Procedures: Definition window. Select **Manage crowd incident** and click **Submit for Approval** (see Figure 5-74 on page 188).

The SOP draft must now go through a formal approval process.

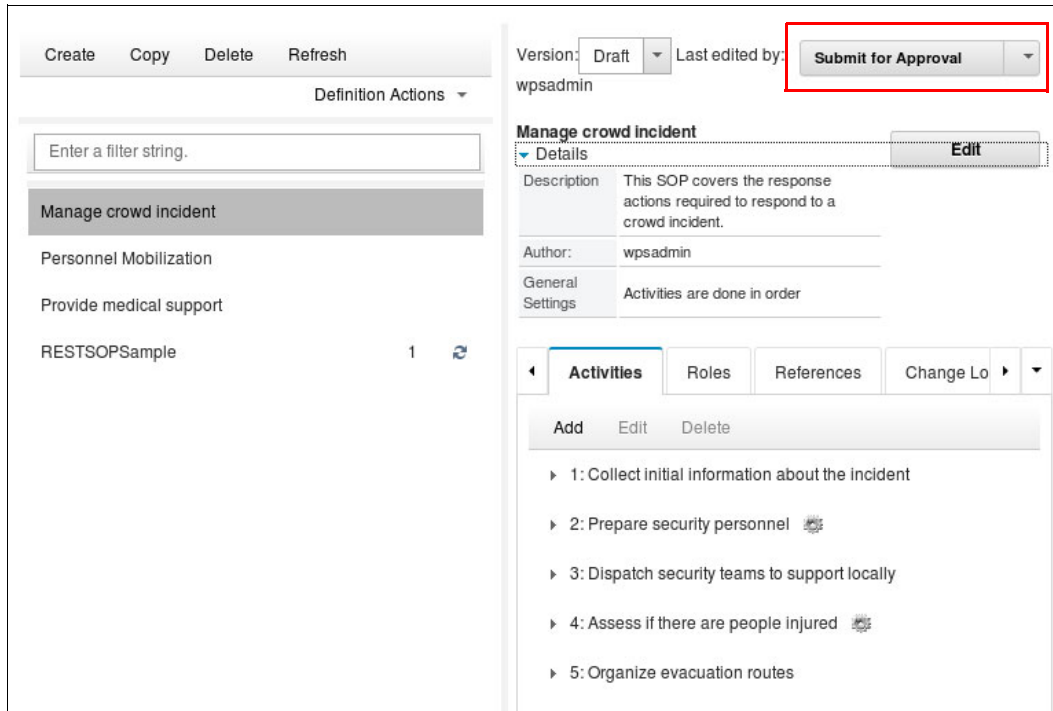


Figure 5-74 Submitting Manage crowd incident SOP draft for approval

3. Approve the Manage crowd incident SOP.

This task must be performed by users with the Owners role in the SOP definition.

The Manage crowd incident SOP definition is listed in the Standard Operating Procedures: Definition window. Select the **Manage crowd incident** and click **Approve**.

The Manage crowd incident SOP is now activated and can be launched (see Figure 5-75).

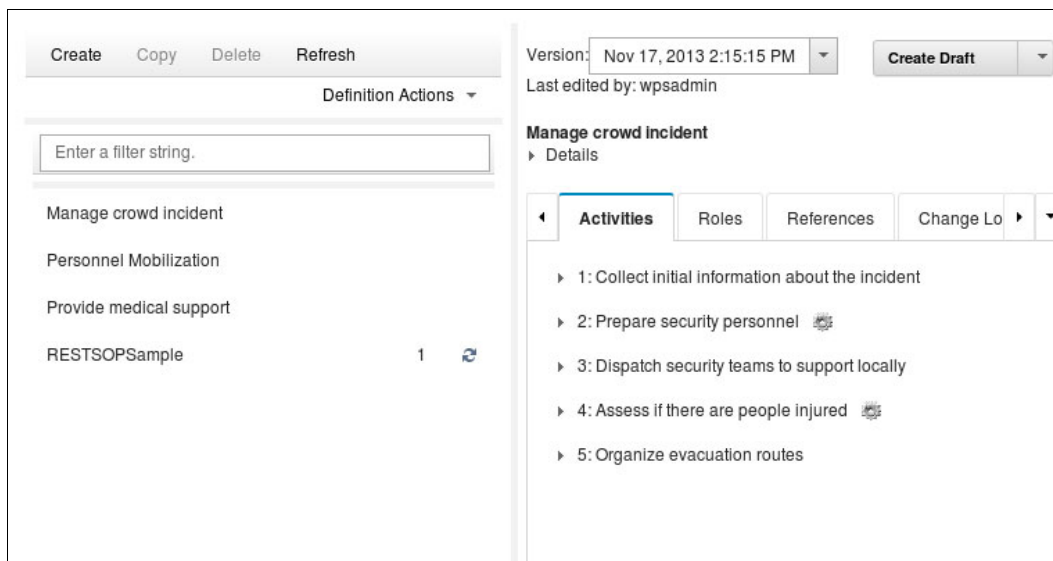


Figure 5-75 Manage crowd incident SOP ready for use

Launching the SOP with a data source routing expression

The Manage crowd incident SOP is started when a crowd incident is created. To associate this SOP with crowd incidents, create a routing expression in the Crowd Gathering data source.

For information about creating data sources, see Chapter 3, “Integrating with data sources” on page 45.

Follow these steps to add a routing expression to the Crowd Gathering data source:

1. Access the Data Sources configuration tool. Select **Solution Administration** → **Configuration Tools** → **Data Sources**.
2. Select the **Crowd Gathering** data source.

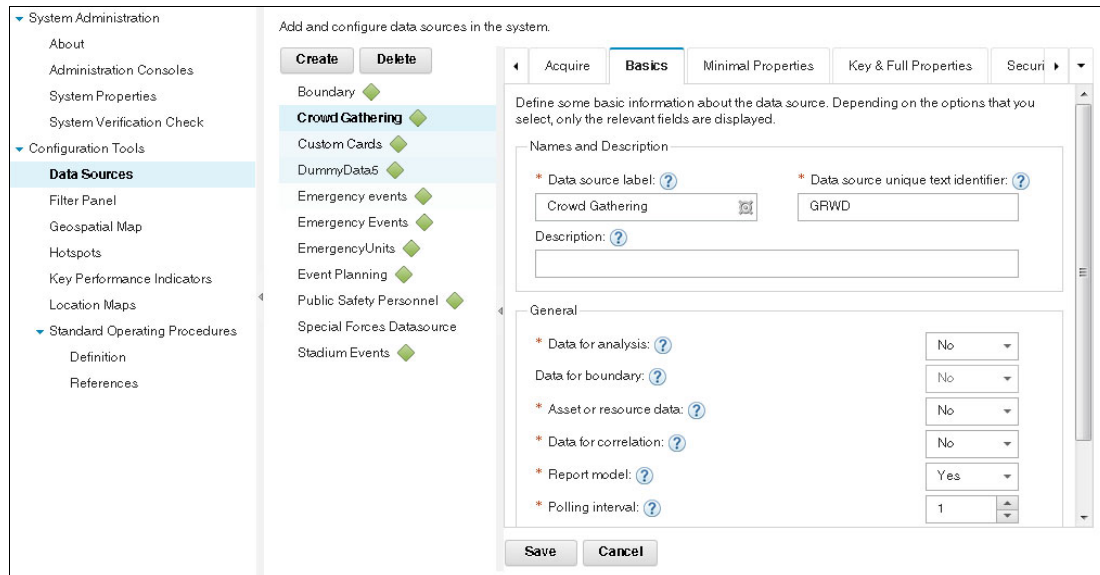


Figure 5-76 Crowd Gathering data source

3. On the Routing tab, click **Add Routing Expression**.
4. Add a Routing expression to route to an SOP when the crowd size (CRWDSIZE) is greater than 20. Select the Manage crowd incident SOP (see Figure 5-77).

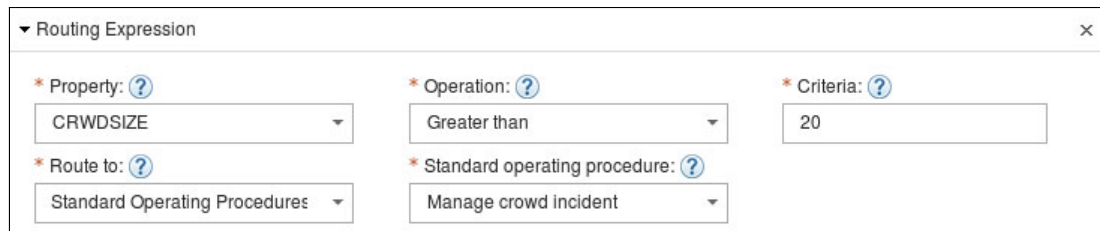


Figure 5-77 Routing expression to launch Manage crowd incident SOP

When the data source data items are imported into IBM Intelligent Operations Center and the crowd size condition in the routing expression is met, the Manage crowd incident SOP is automatically started.



Customizing the operations display

This chapter describes the main customization options available for the user interface of IBM Intelligent Operations Center.

This chapter includes the following topics:

- ▶ 6.1, “User interface customization overview” on page 192
- ▶ 6.2, “Customizing the content panel” on page 197
- ▶ 6.3, “Creating custom data source actions” on page 219
- ▶ 6.4, “Adding filter-driven result charts” on page 227
- ▶ 6.5, “Customizing preview cards” on page 233
- ▶ 6.6, “Creating custom portlets” on page 239
- ▶ 6.7, “Adding custom help links” on page 252
- ▶ 6.8, “Customizing the More Actions menu” on page 255
- ▶ 6.9, “Creating custom solution pages” on page 257

Note: For information about the sample code for the examples described in this chapter, see Appendix B, “Samples provided with this book” on page 383.

6.1 User interface customization overview

The user interface of IBM Intelligent Operations Center is based on IBM WebSphere Portal Version 8.0 and provides several customization options. However, some of the options require additional licenses for the development environment, such as IBM Rational Application Developer. This section provides a brief review of each customization option.

6.1.1 Content panel customization

By default, the content panel has three tabs: Map, Location map, and List. Through customization, you can change the order of these tabs and add or remove custom tabs. Figure 6-1 shows an example of a custom tab.

This approach can be used to create custom representations of data from the data sources or to integrate with other systems using Dojo widgets. For details, see 6.2, “Customizing the content panel” on page 197.

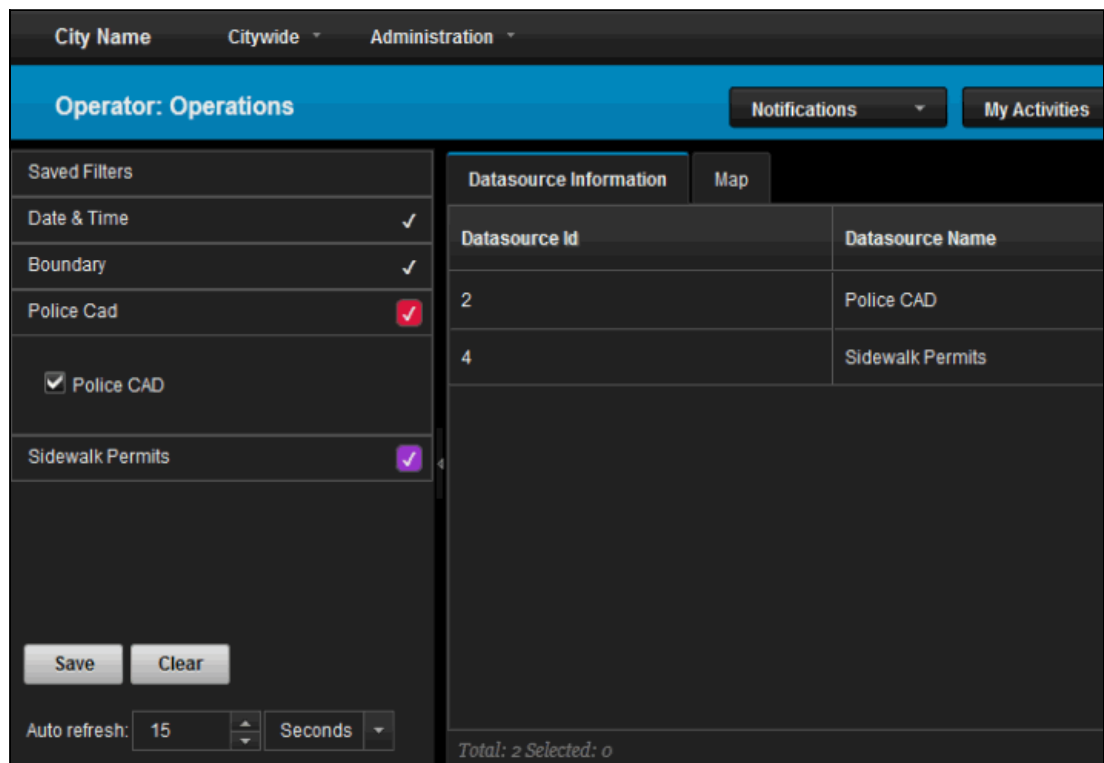


Figure 6-1 Custom tab in custom viewer portlet showing data source IDs and names

6.1.2 Creating custom data source actions

Custom data source actions can be created for each data source. This customization provides the capability to add new actions to the *More Actions* menu in the preview card for all associated data items. Figure 6-2 on page 193 shows an example of a custom data source action in the preview card *More Actions* menu.

Such an action can provide the function to integrate with a third-party system, or to extend existing product functions. For details, see 6.3, “Creating custom data source actions” on page 219.

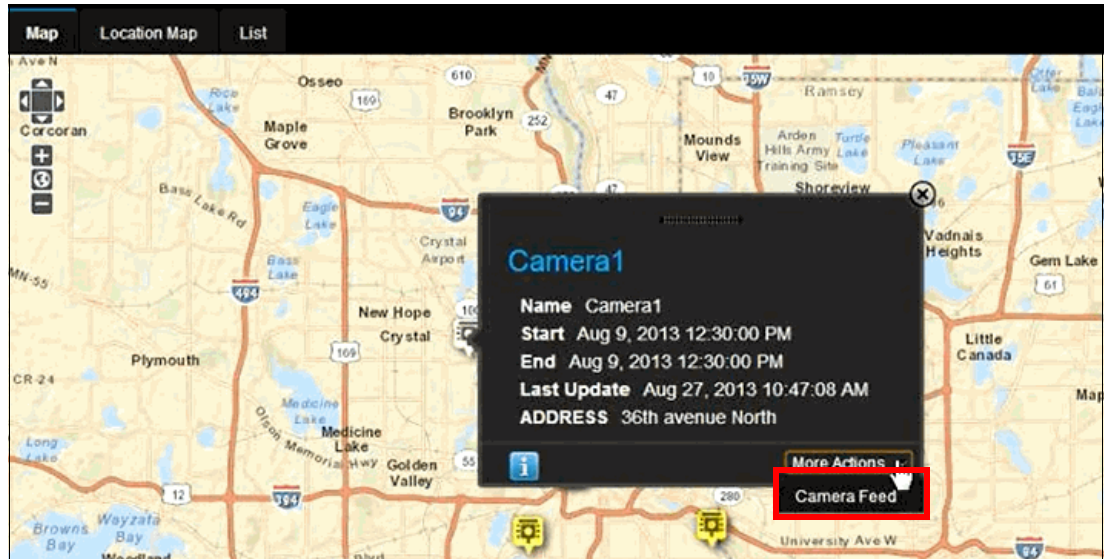


Figure 6-2 Custom data source action in the More Actions menu

6.1.3 Adding custom filter-driven results charts

The default charts on the right supporting content panel are filter-driven and show statistics for currently filtered data from data sources. These charts are built using Dojo charting widgets. This customization option allows you to add new Dojo widgets and provide alternative data visualization. Figure 6-3 on page 194 shows an example of a custom Dojo chart.

For details, see 6.4, “Adding filter-driven result charts” on page 227.

For more information about Dojo visualization, see *Graphics, Charting and Visualization*:

<http://dojotoolkit.org/features/graphics-and-charting>

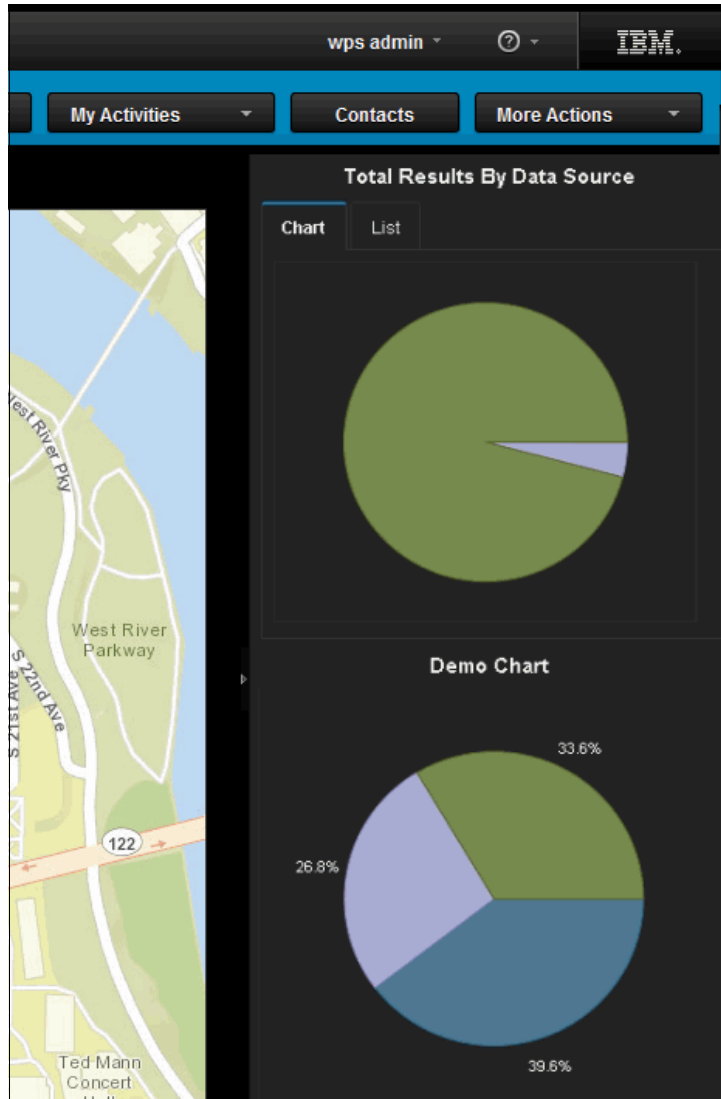


Figure 6-3 Custom chart on right supporting content panel

6.1.4 Customizing preview cards

By default, all the properties for data items in preview cards are in text format. However, it is possible to add other types of information, such as images, links, or even video, as shown in Figure 6-4 on page 195.

For details, see 6.5, “Customizing preview cards” on page 233.

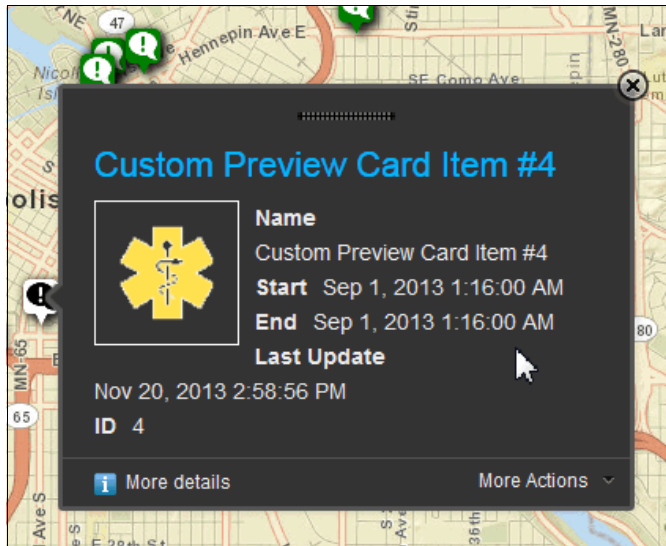


Figure 6-4 Custom preview card

6.1.5 Creating custom portlets

The user interface of IBM Intelligent Operations Center can be extended by custom portlets. A developer can use the following services to optimize the user experience and performance:

- ▶ RestStore JavaScript class for accessing Representational State Transfer (REST) services

The RestStore JavaScript class is based on Dojo JavaScript Library. It enables access to REST services data from the local data store instead of calling it directly. This approach reduces the number of back-end calls and reuses the data stored locally, which makes the user interface more responsive and decreases the load on the server and network.

- ▶ Translation service

The translation service is used to store user interface labels and other displayed text in different languages. It renders the text in the appropriate language based on the preferred user locale.

For details, see 6.6, “Creating custom portlets” on page 239.

6.1.6 Adding custom help links

This feature is useful when you need to provide documentation for a new custom feature and make it easily accessible from the standard Help menu, as shown in Figure 6-5 on page 196.

For details, see 6.7, “Adding custom help links” on page 252.

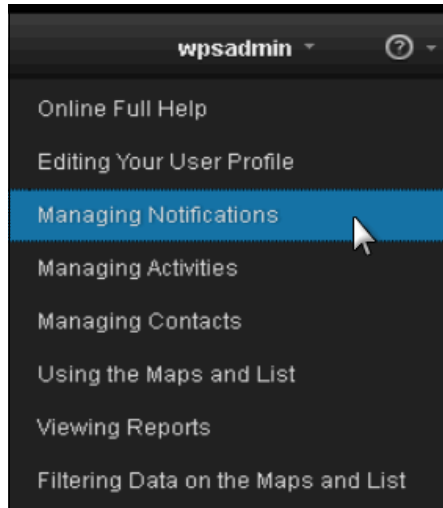


Figure 6-5 Link to a custom help topic in the standard Help menu

6.1.7 Customizing the More Actions menu

The More Actions menu on the upper-right corner of the Operations view allows you to add or remove new options using custom JavaScript code. For example, it is possible to call a custom web form or add data into a custom portlet or widget. Figure 6-6 shows an example of a custom More Actions menu option.

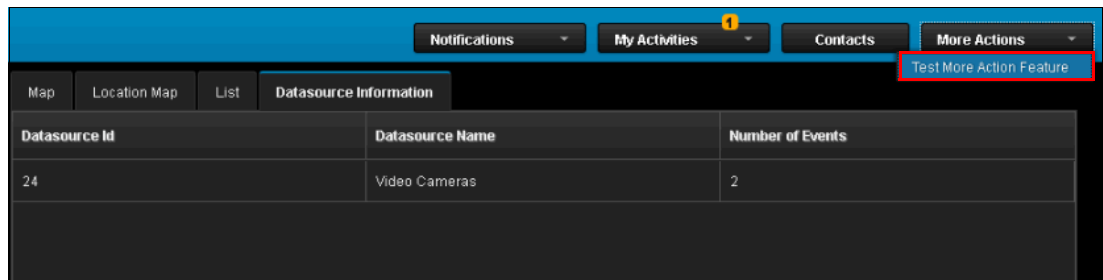


Figure 6-6 Custom More Actions menu option

For details, see 6.8, “Customizing the More Actions menu” on page 255.

6.1.8 Creating custom portal pages

Creating new custom portal pages can be useful when creating a new user interface for a new user role or as a way to provide new functions with custom portlets. This is a standard feature that is provided by IBM WebSphere Portal and a portal theme included with IBM Intelligent Operations Center. Figure 6-7 on page 197 shows an example of a portal page with the standard IBM Intelligent Operations Center theme.

For details, see 6.9, “Creating custom solution pages” on page 257.

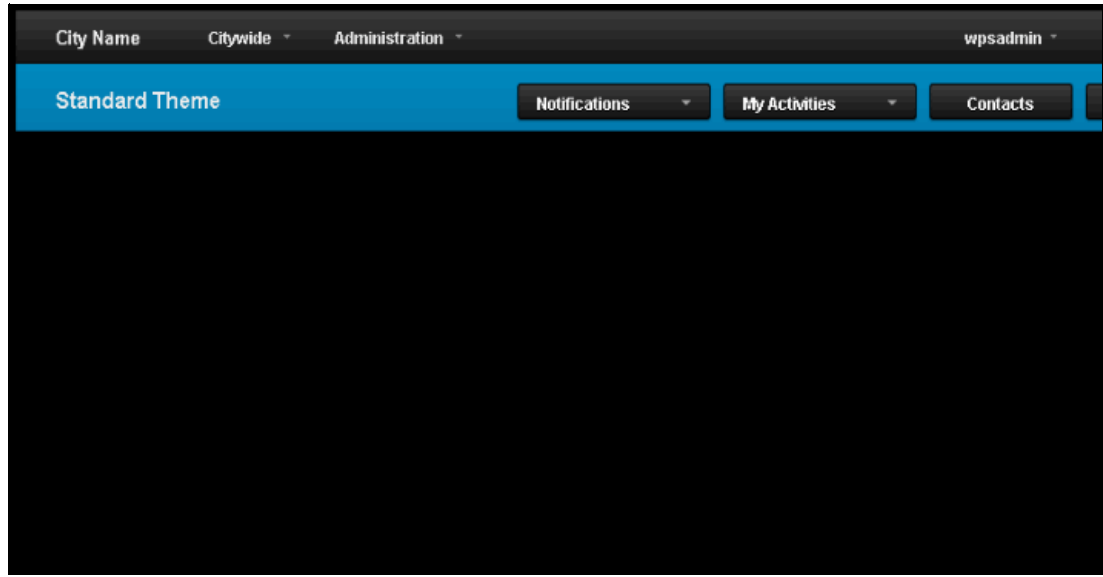


Figure 6-7 Custom portal page with standard IBM Intelligent Operations Center portal theme

6.2 Customizing the content panel

The content panel is implemented through the content viewer portlet, which allows you to create new tabs and load custom Dojo widgets inside them.

Customizing the content panel is useful for integrating with external systems with a web user interface or to use a custom widget that will interact with or represent data from the REST services in alternative ways.

The following examples are more practical examples of content panel customization:

- ▶ List additional properties from the data items that are not listed on the List tab.
- ▶ Integrate with an external website page using a custom Dojo widget that can interact with both a back-end system and IBM Intelligent Operations Center REST services.
- ▶ Show the web page from a back-end system through the `iframe` HTML tag within the Dojo widget.

6.2.1 Implementation summary

The following summary steps extend the content viewer portlet:

1. Create a custom widget.
2. Generate an enterprise archive (EAR) file with custom web application code and deploy it to IBM WebSphere Application Server.
3. Make a copy of the standard content viewer portlet and configure it to load a new widget from a previously deployed custom application.
4. Create and configure a new portal page and add the new portlet to the page.

The following two samples are used in this section:

- ▶ `sample_ioc_contentviewer_web`
- ▶ `sample_ioc_contentviewer_web_ear`

For more information about the samples included with this book, see Appendix B, “Samples provided with this book” on page 383.

6.2.2 Setting up the development environment

This example requires you to set up IBM Rational Application Developer V8.5. For information about setting up IBM Rational Application Developer, see “Setting up IBM Rational Application Developer for WebSphere Software 8.5” on page 379.

IBM Rational Application Developer is used to develop and package the custom action Dojo module.

Note: In this example, IBM Rational Application Developer is used to generate the EAR file from the sample code provided with this book. For instructions about how to configure IBM Rational Application Developer for user interface customization development, see 6.6.1, “Custom portlet development environment” on page 239.

6.2.3 Implementation details

This section includes step-by-step instructions to customize the content panel in this example.

Creating a custom widget for the content panel

Follow these steps:

1. Extend the default widget `ioc/contentviewer/View`.

The standard user interface of IBM Intelligent Operations Center is built using the Dojo framework and can be extended with standard Dojo extension capabilities, such as widgets. To create a custom widget for the content panel, extend the default widget `ioc/contentviewer/View` by using the `declare` function from the `dojo/_base/declare` module, as shown in Example 6-1.

Note: The `ioc/contentviewer/View` widget is defined in the IBM Intelligent Operations Center portal theme file:

```
/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_theme_ear.ear/ioc_theme_static_js_web.war/js/com/ibm/ioc/contentviewer/View.js
```

Example 6-1 Extending standard `ioc/contentviewer/View` widget with `declare` function

//For more details, see
<http://dojotoolkit.org/documentation/tutorials/1.7/declare/>.

```
define(  
  [  
    "dojo/_base/declare",  
    "dojo/_base/lang",  
    "dojo/when",  
    "ioc/com/ibm/ioc/contentviewer/View",  
    "dijit/_TemplatedMixin",  
  
    // [...]
```



```

    ],
    function(declare, lang, when, View, _TemplatedMixin, _WidgetsInTemplateMixin,
    RestStore, SessionData, JSLogger, Library, template, css) {

    return declare("test.contentviewerWidget", [View, _TemplatedMixin,
    _WidgetsInTemplateMixin], {

    // [...]

    });

```

2. Make a note of the value used in the declare function for the className argument.

In Example 8-1, this value is `test.contentviewerWidget`. It will be needed later for defining the value for the **MODULE** parameter in the content viewer configuration. For more details, see “Adding a new tab with a custom widget into the portlet configuration” on page 217.

3. Set the title of the tab in the constructor as shown in Example 6-2.

This title will be displayed on the tab, for example, Data Source Information Custom. The internationalization service can be used to translate the title into other languages. For more information about the internationalization service, see 6.6.4, “Using the translation widget and internationalization service” on page 248.

Example 6-2 Defining the title for a new tab in the constructor

```

constructor: function(/*Object*/ kwArgs) {
    this._logger = new JSLogger({name: "contentViewerWidget"});
    var methodName = "constructor";
    this._logger.traceEntry(methodName);
    lang.mixin(this, kwArgs);
    this.title = "Datasource Information";
    this._loadedDatasourcesCache = [];
    this._library = new Library();
    this._logger.traceExit(methodName);
},

```

4. Implement the functions listed in Table 6-1 on page 200.

If there are any required interactions with the data shared across all the views, such as Location Map, Map, List, and the charts, implement the functions listed in Table 6-1 on page 200. Using these functions, you can connect to a data source, update the data source criteria, toggle data source on and off, and customize the Help and More Actions menus.

Table 6-1 Functions defined in ioc/contentviewer/View widget that can be extended

Function name	Function parameters	Description
<pre>addDatasource: function(datasourceConfig, datasourceStore, filterDatasourceInfo){ };</pre>	<p>datasourceConfig - Object containing data source properties</p> <p>datasourceStore - {Dojo observable store}</p> <p>filterDatasourceInfo - Information passed from the filter about the data source</p>	<p>Called when a data source is first selected in the filter.</p> <p>For more information, see Example 6-7 on page 227.</p>
<pre>updateDatasourceCriteria: function(datasourceConfig, datasourceStore, filterDatasourceInfo){ },</pre>	<p>datasourceConfig - Object containing data source properties</p> <p>datasourceStore - {Dojo observable store}</p> <p>filterDatasourceInfo - Information passed from the filter about the data source</p>	<p>Called when a data source criteria or color has been changed.</p> <p>For an example, see the source code of the standard filter-driven charts, available in /opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_portal_ear.ear/ioc_ui_charting_portlet.war/_ioc_ui_charting_portlet/js/ChartView.js</p>
<pre>removeDatasource: function(datasourceId){ },</pre>	<p>datasourceId - {Integer} The identifier of the data source</p>	<p>Called when a data source has been removed.</p> <p>For an example, see the source code of the standard filter-driven charts, available in /opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_portal_ear.ear/ioc_ui_charting_portlet.war/_ioc_ui_charting_portlet/js/ChartView.js</p>
<pre>toggleDatasource: function(datasourceId, displayFlag){ },</pre>	<p>datasourceId - {Integer} The identifier of the data source</p> <p>displayFlag - {boolean} flag indicating whether the data source needs to be displayed</p>	<p>Called when a loaded data source has been toggled on or off.</p> <p>For an example, see the source code of the standard filter-driven charts, available in /opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_portal_ear.ear/ioc_ui_charting_portlet.war/_ioc_ui_charting_portlet/js/ChartView.js</p>
<pre>showHelp: function(){ },</pre>	<p>None</p>	<p>Called in the scope of the View when the user clicks the Help link in the More Actions menu.</p> <p>For more information, see 6.7, “Adding custom help links” on page 252.</p>

Function name	Function parameters	Description
<pre>getMoreActionsMenuItems: function() { },</pre>	<p>None</p> <p>This function returns an object with string key and value of {label, callback} or a promise that will be resolved to that object</p>	<p>Retrieves the set of actions that will be used to populate the More Actions drop-down menu. For more information, see 6.8, “Customizing the More Actions menu” on page 255.</p>

For more examples, see the `contentviewerWidget.js` file inside the `sample_ioc_contentviewer_web` sample that is provided with this book.

Generating and deploying the EAR file

In the example provided with this book, the EAR file is a custom Java Platform, Enterprise Edition application and contains a web application code.

Using the approach demonstrated in this sample, developers can access the REST APIs or integrate with third-party applications through the user interface using standard web development technologies, such as Dojo code.

The following sections describe how to import the sample code provided with this book into IBM Rational Application Developer and generate a deployable EAR file. The sample code that is imported into the Rational Application Developer can be modified later to implement other features.

Importing the sample code into IBM Rational Application Developer

Follow these steps:

1. Start Rational Application Developer and click **File** → **Import** (see Figure 6-8).

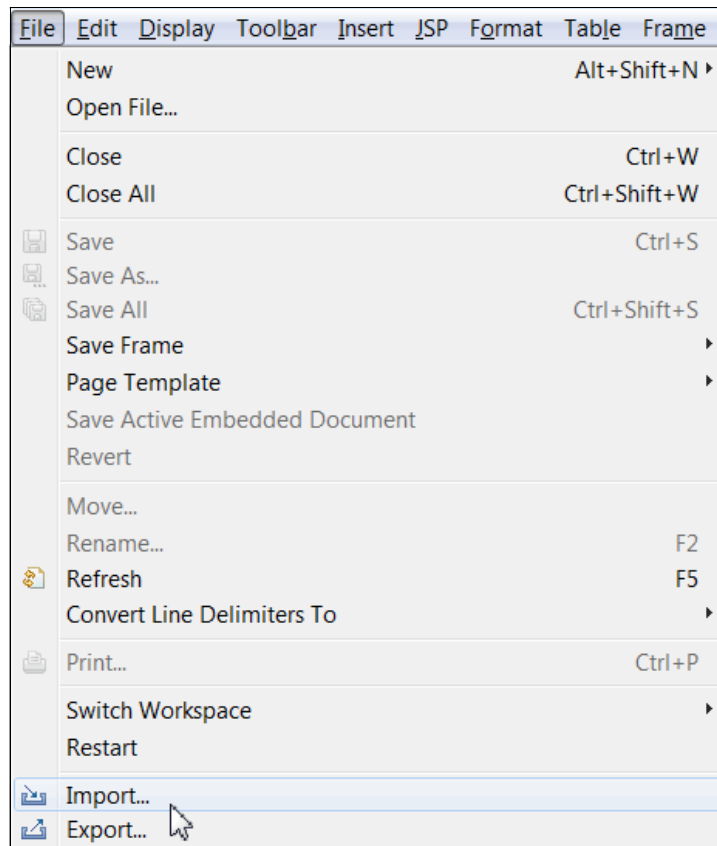


Figure 6-8 Preparing to import the sample code into IBM Rational Application Developer workspace

2. Click **General** → **Existing Projects into Workspace** and click **Next** (see Figure 6-9).

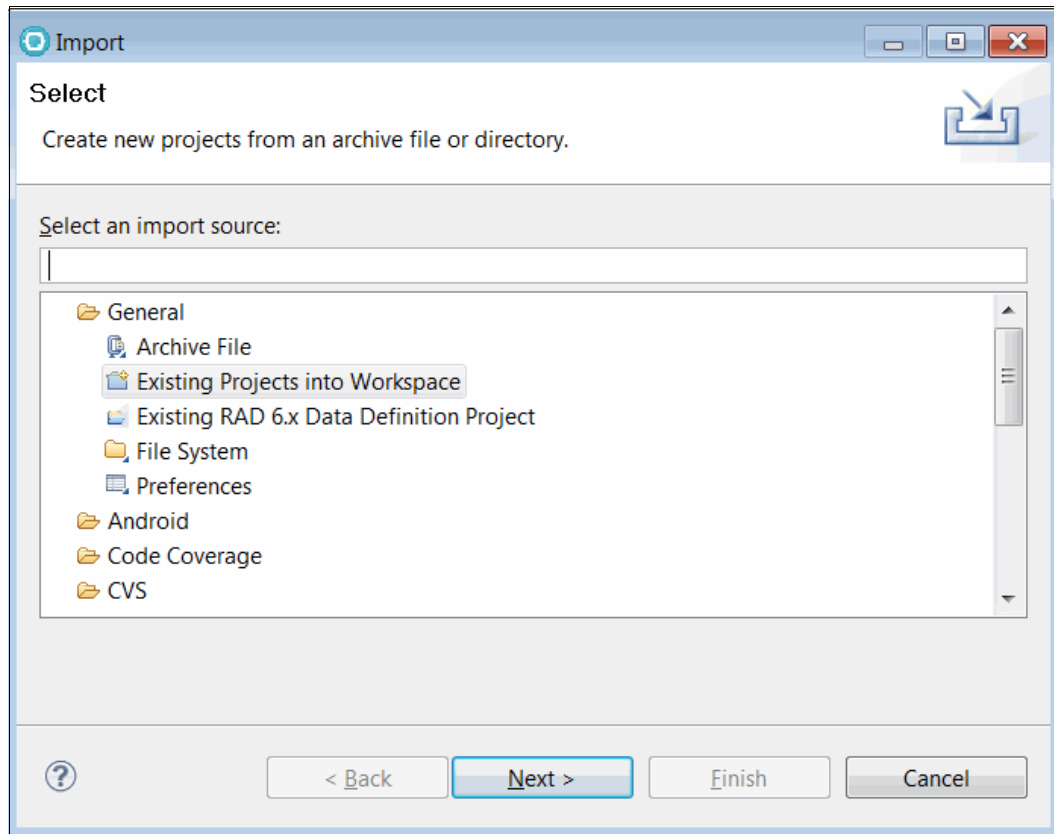


Figure 6-9 Selecting to import existing projects into workspace

3. Click **Select root directory** and click **Browse** (see Figure 6-10):
 - a. Navigate to the folder with the samples.
 - b. Select **sample_ioc_contentviewer_web** and **sample_ioc_contentviewer_web_ear**.
 - c. Click **Finish**.

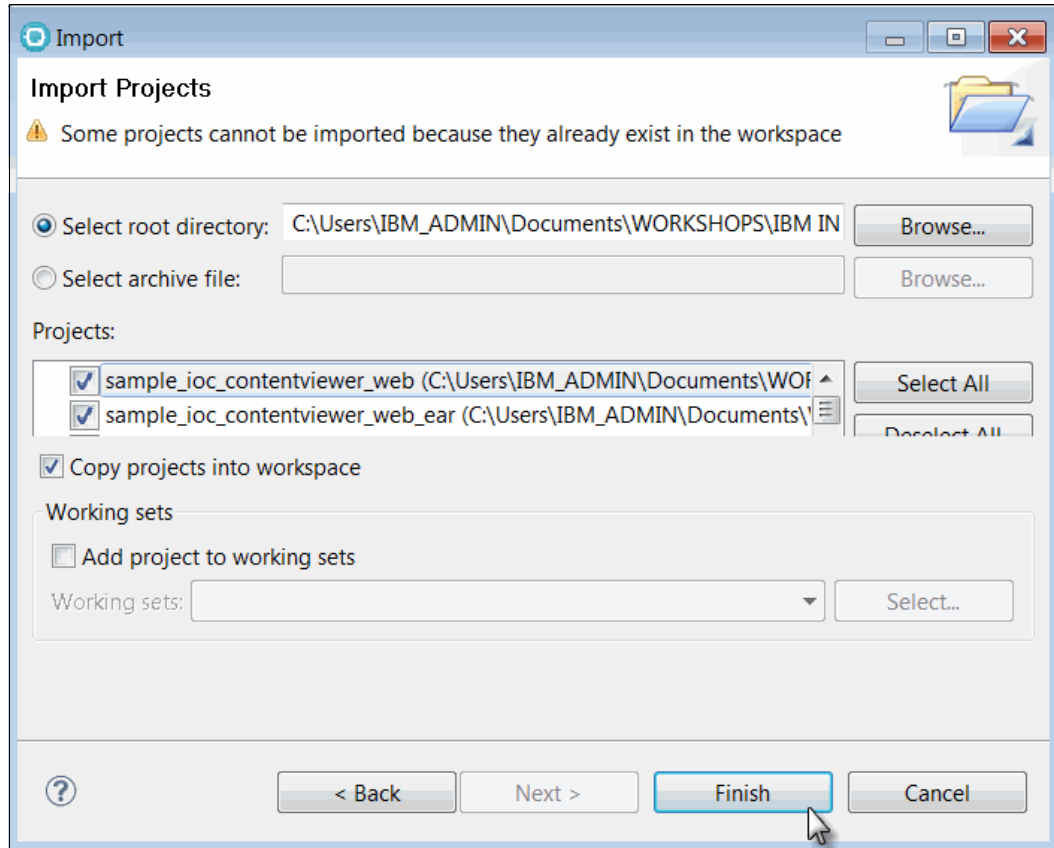


Figure 6-10 Importing existing samples into workspace

Exporting the EAR file

Follow these steps:

1. On the Project Explorer tab, right-click **sample_ioc_contentviewer_web_ear** and select **Export** → **EAR file** (see Figure 6-11).

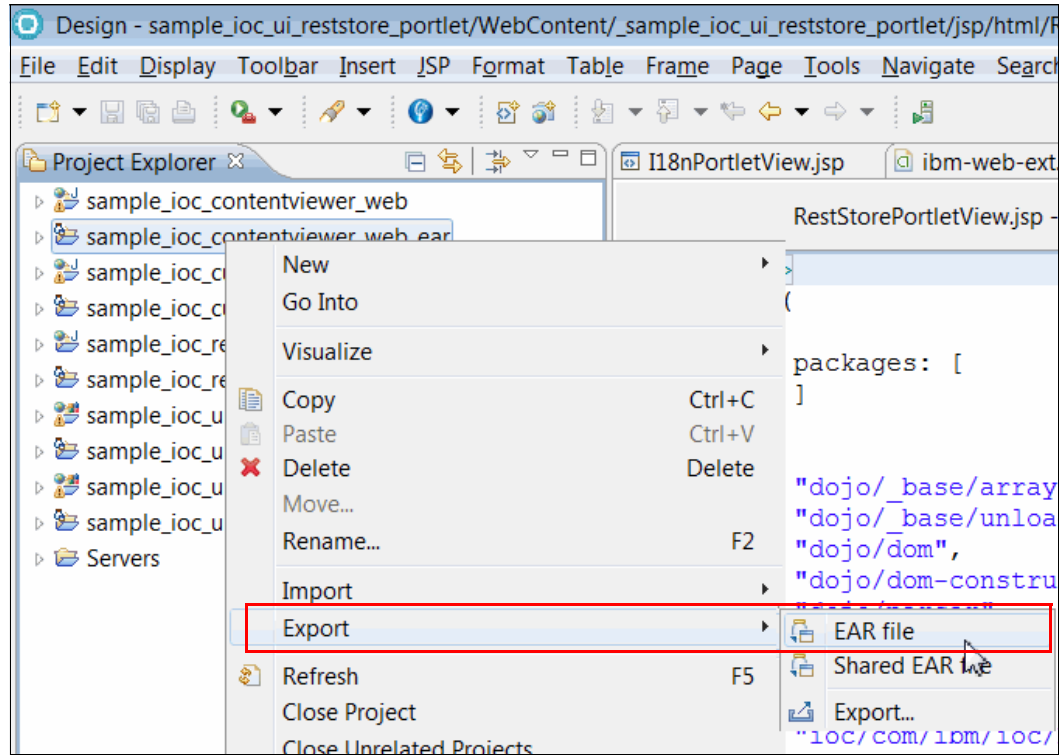


Figure 6-11 Preparing to export the EAR file

2. Click **Browse** next to the Destination field and select the path to the folder where the exported file will be placed. Click **Finish** (see Figure 6-12).

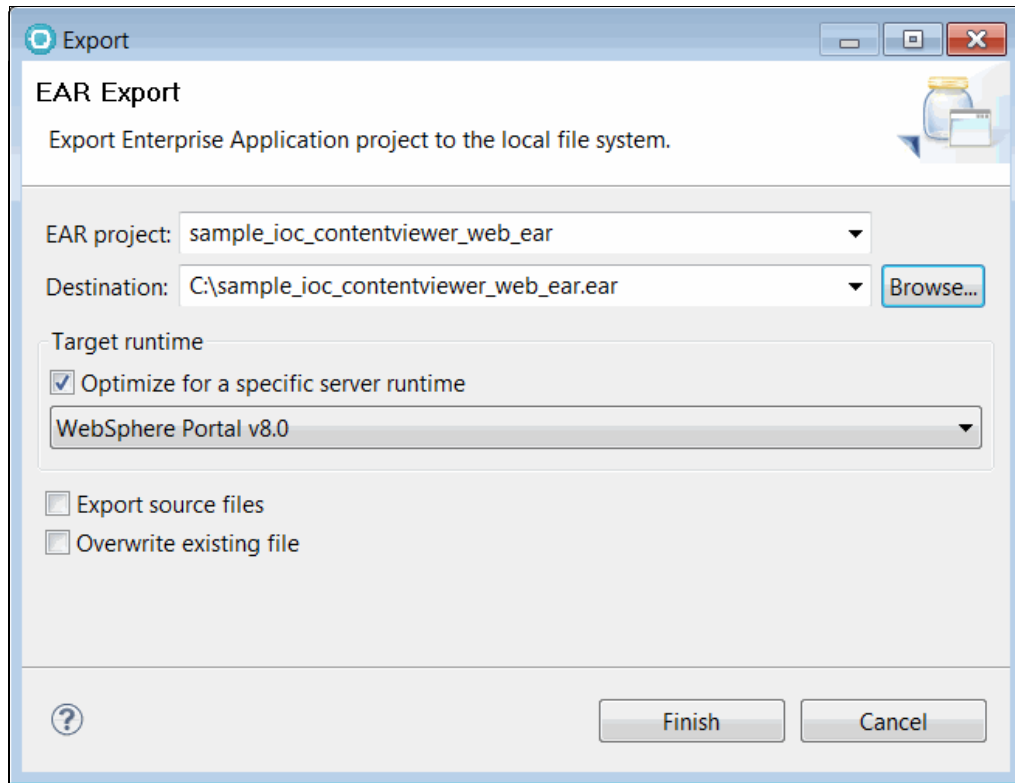


Figure 6-12 Exporting the EAR file to the file system

Installing the new application in IBM WebSphere Application Server

Follow these steps:

1. Log in to the WebSphere Application Server 8.0 Deployment Manager console on the application server with the administrative account. You can access the console from the IBM Intelligent Operations Center portal:

Click **Administration** → **Solution Administration** → **System Administration** → **Administration Consoles** → **Application server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Sametime Proxy)

Alternatively, you can access the console with the URL:

`https://<ioc_app>:9044/ibm/console/`

2. Click **Applications** → **New Application** (see Figure 6-13).

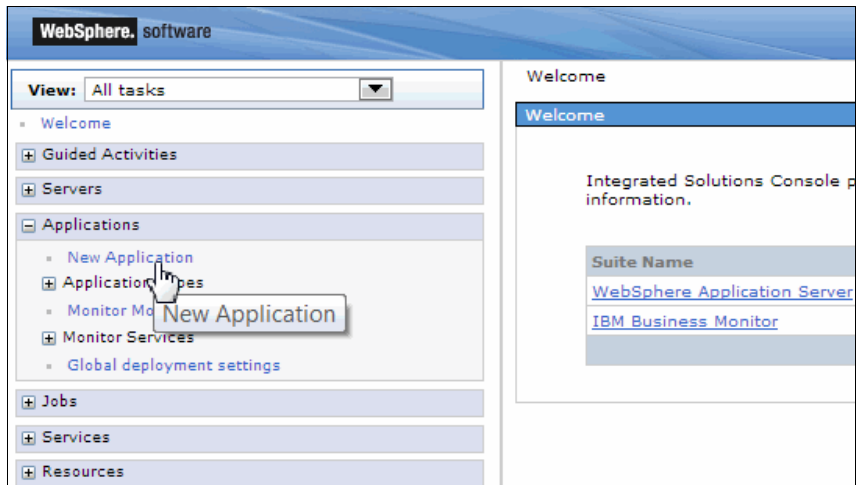


Figure 6-13 Navigating to the New Application wizard

3. Click **New Enterprise Application** (see Figure 6-14).

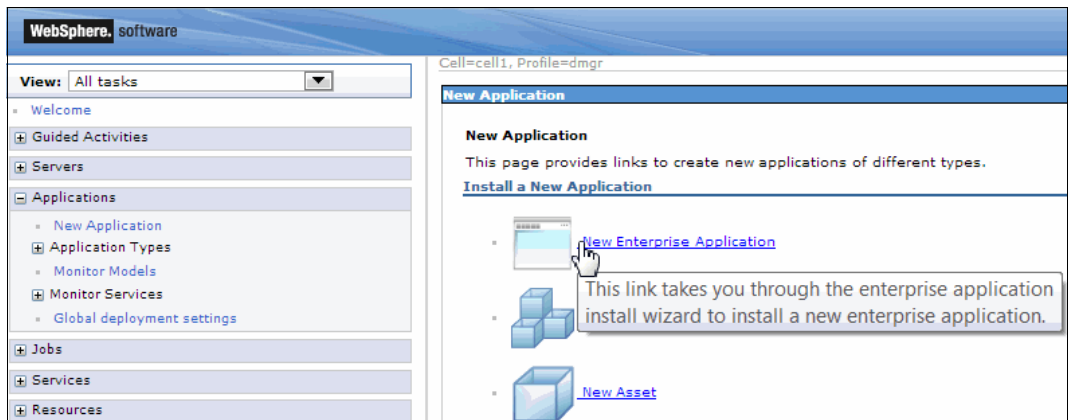


Figure 6-14 Choose to install New Enterprise Application

4. In the Path to the new application section, select **Local file system**, click **Choose File** and navigate to the EAR file exported in “Exporting the EAR file” on page 205. Click **Next** (see Figure 6-15).

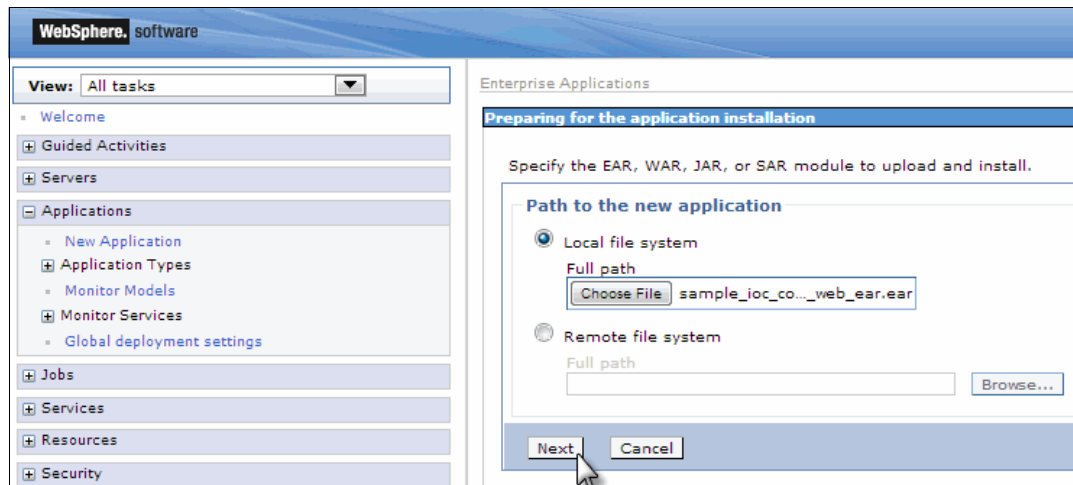


Figure 6-15 Uploading the EAR file to the server

5. Select **Fast Path - Prompt only when additional information is required** and click **Next** (see Figure 6-16).

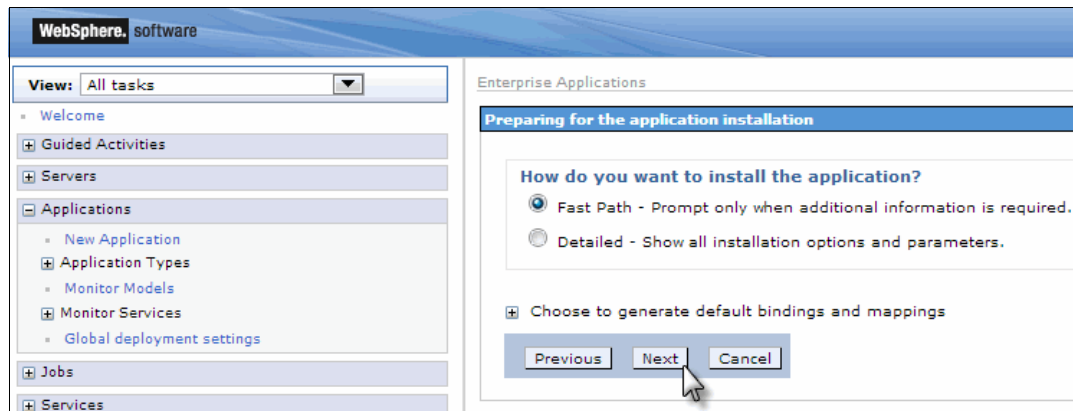


Figure 6-16 Selecting Fast Path for application deployment

6. Accept the defaults for all fields in Step 1: Select installation options and click **Next**.

7. On Step 2: Map modules to servers (see Figure 6-17), follow these steps:
 - a. Select the **sample_ioc_contentviewer_web.war** module.
 - b. From the list under Clusters and servers, select the following targets:
 - WebSphere:cell=cell1,node=ihsnode1,server=ihsserver1**
 - WebSphere:cell=cell1,cluster=PortalCluster**
 - c. Click **Apply** and then click **Next**.

The web module inside the Java Platform, Enterprise Edition application is deployed to the WebSphere Portal Server, and it is accessible through the HTTP server.

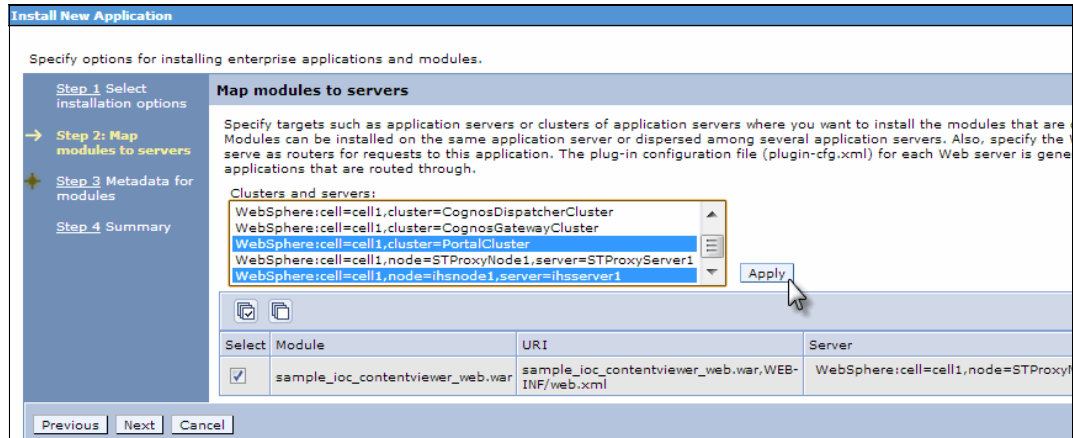


Figure 6-17 Mapping modules to servers

8. Accept the default values in Step 3: Metadata for modules, and click **Next**.
9. On Step 4: Summary, click **Finish**.
10. Click **Save** (see Figure 6-18).

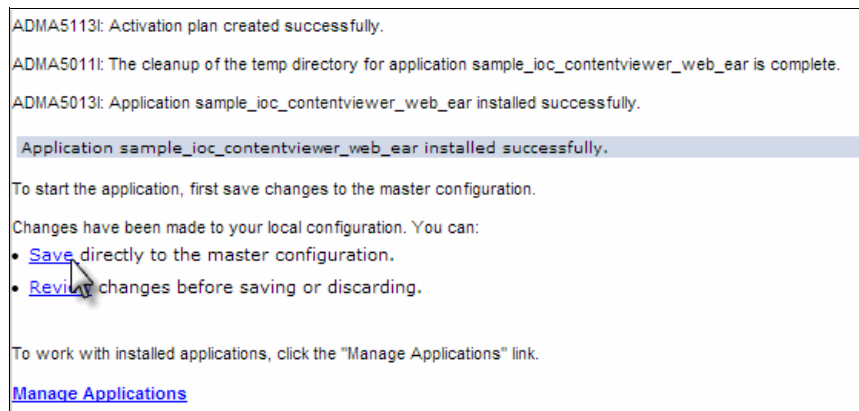


Figure 6-18 Saving the configuration

11. Click **Save directly to the master configuration** (see Figure 6-19).

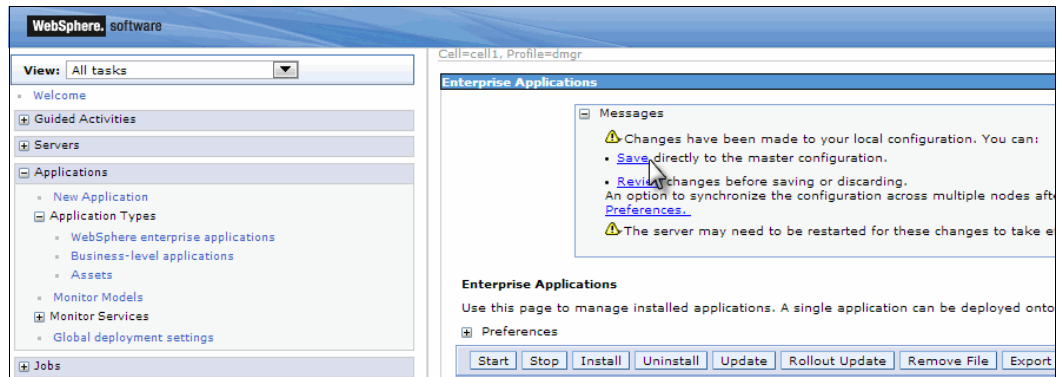


Figure 6-19 Saving changes to master configuration

Starting and testing the new application

Follow these steps:

1. Navigate to **Applications** → **Application Types** → **WebSphere enterprise applications** (see Figure 6-20).



Figure 6-20 Navigating to the list of installed applications

2. Search for **sample_ioc_contentviewer_web_ear**, select it by clicking the check box next to the application name, and click **Start** (see Figure 6-21).

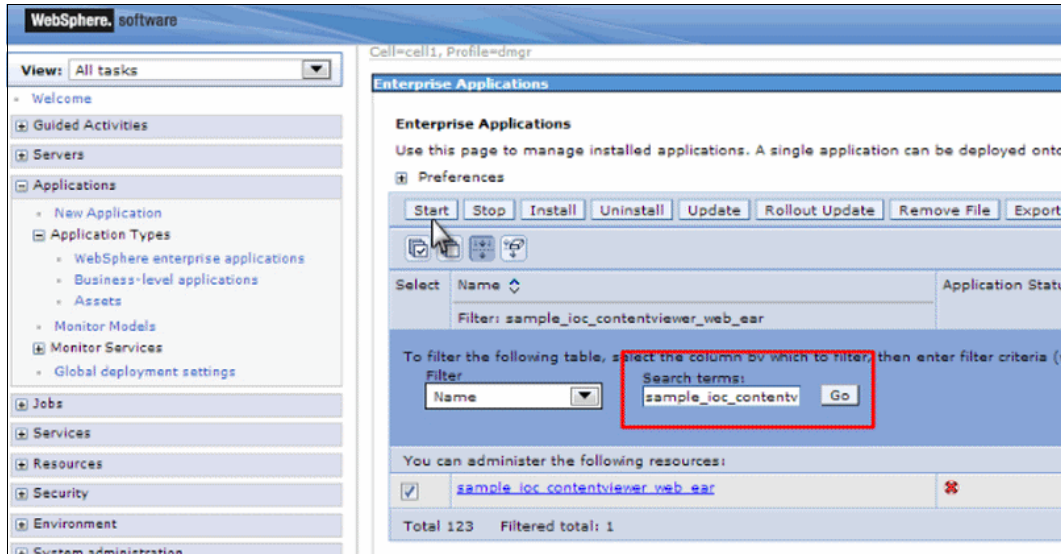


Figure 6-21 Starting the application

3. Test the application.

Open a new tab in the browser and enter the URL:

`https://<ioc_ibs>/sample/custom/contentviewer/js/test/contentviewerWidget.js`

The content of the JavaScript file is displayed on the browser as shown in Figure 6-22.

```

/*
 * Licensed Materials - Property of IBM
 *
 * 5725-D69
 *
 * (C) Copyright IBM Corp. 2012 All rights reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */

define(
  [
    "dojo/_base/declare",
    "dojo/_base/lang",
    "dojo/when",
    "ioc/com/ibm/ioc/contentviewer/View",
    "dijit/_TemplatedMixin",
    "dijit/_WidgetsInTemplateMixin",
    "dojo/store/Memory",
    "dojo/store/Observable"
  ],
  function(declare, lang, when, View, TemplatedMixin, WidgetsInTemplateMixin, Memory, Observable) {
    // ...
  }
);

```

Figure 6-22 Testing the application - Expected output

Copying the standard content viewer portlet and modifying the configuration

There are many ways to change the configuration of the portlet in WebSphere Application Server. In this example, a new copy of the portlet is created to isolate the custom configuration from all other portlet instances on other pages. But, if the goal is to modify all the instances at one time, modify the original portlet configuration instead.

Creating a copy of the content viewer portlet

Follow these steps to create a copy of the content viewer portlet:

1. Log on to IBM Intelligent Operations Center with an administrator user account and navigate to **Administration** → **Portal Administration** (see Figure 6-23).

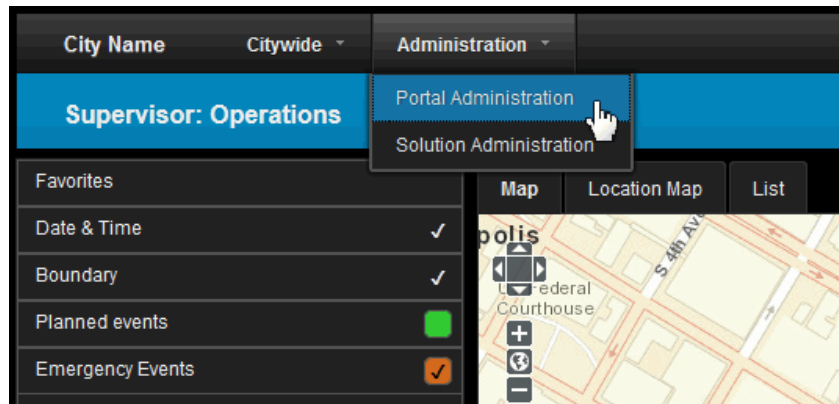


Figure 6-23 Navigating to the Portal Administration interface

2. Navigate to **Portlet Management** → **Portlets** (see Figure 6-24).

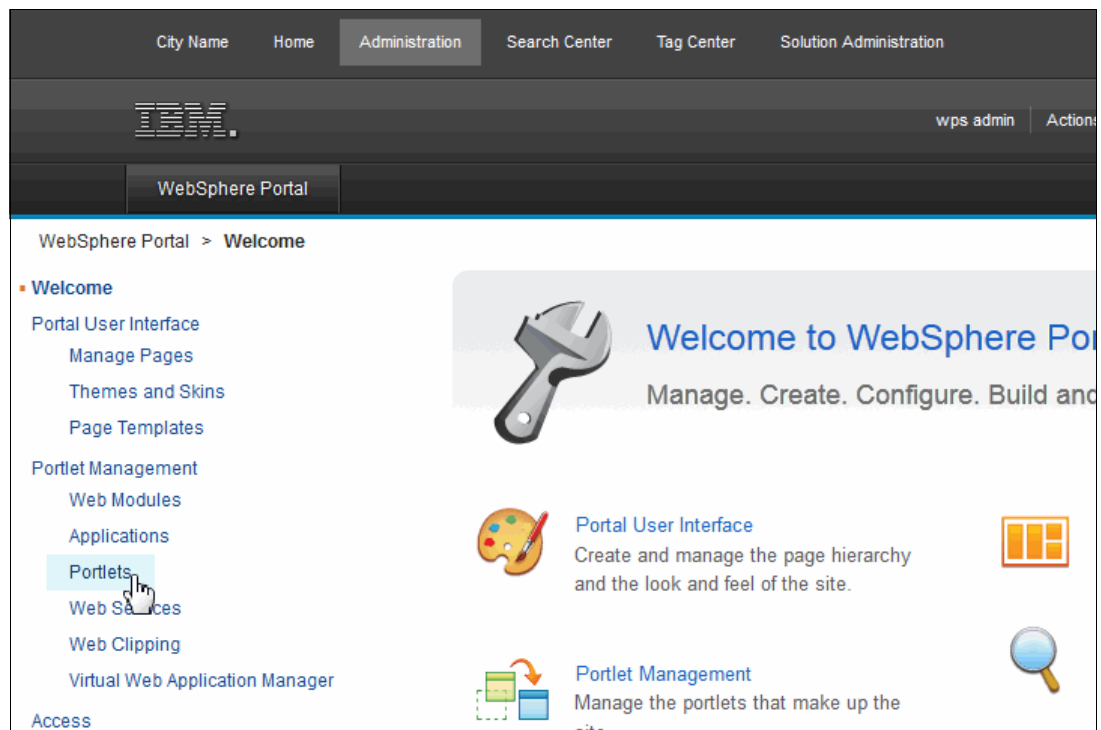


Figure 6-24 Accessing the Portlets administration page

3. In the Search field, enter `ioc_ui_content` and click **Search** (see Figure 6-25).

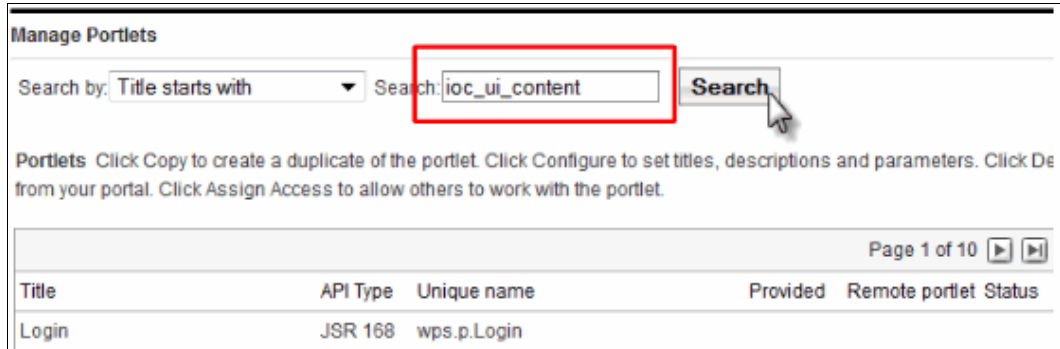


Figure 6-25 Searching for the default content portlet

4. Click the **Copy portlet** icon (highlighted in Figure 6-26).

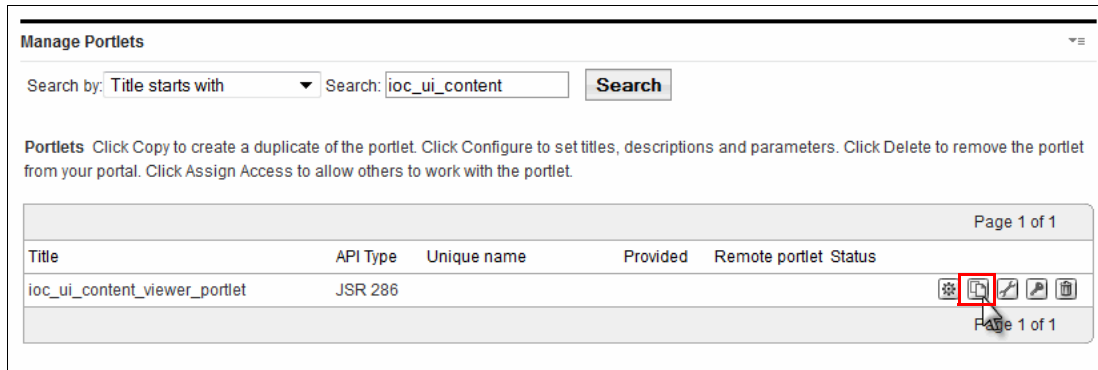


Figure 6-26 Creating a copy of the default content portlet

5. Type a name for the new portlet copy (for example, `ioc_ui_content_viewer_portlet_custom`) and click **OK** (see Figure 6-27).

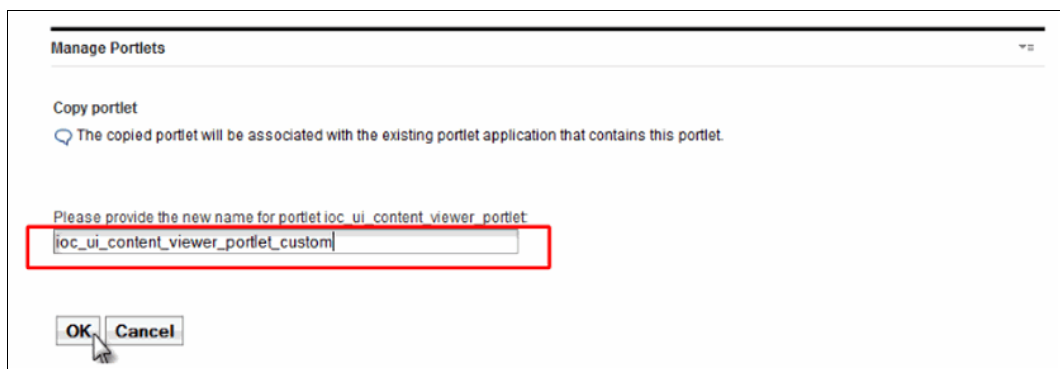


Figure 6-27 Assigning a name to the new portlet copy

Changing the access rights for the new copy of the content viewer portlet

After creating a copy of the content viewer portlet, the administrator must give access to it to all authenticated portal users.

Follow these steps:

1. Click the **Assign access to portlet** icon (highlighted in Figure 6-28).

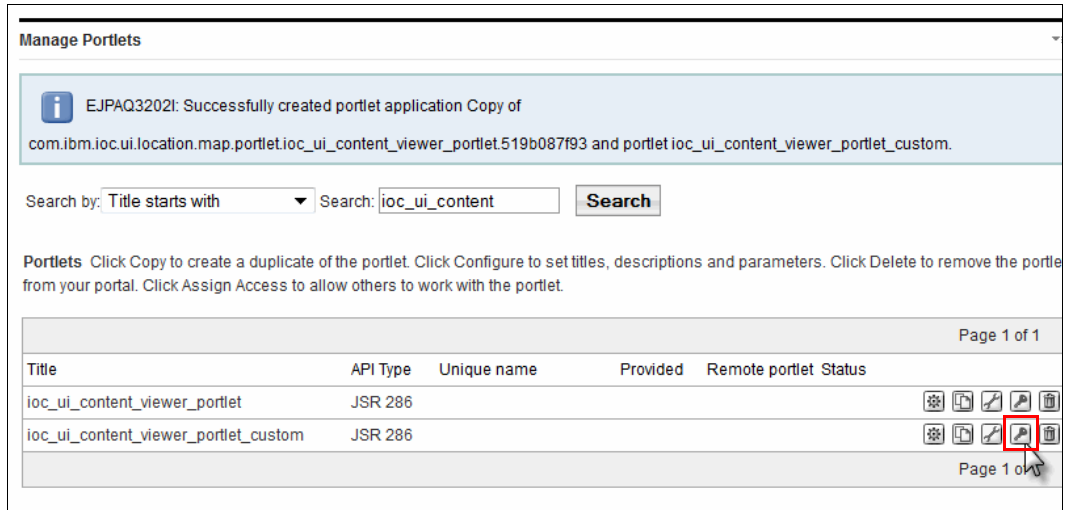


Figure 6-28 Changing access rights

2. For the User role, click the **Edit role** icon (see Figure 6-29).

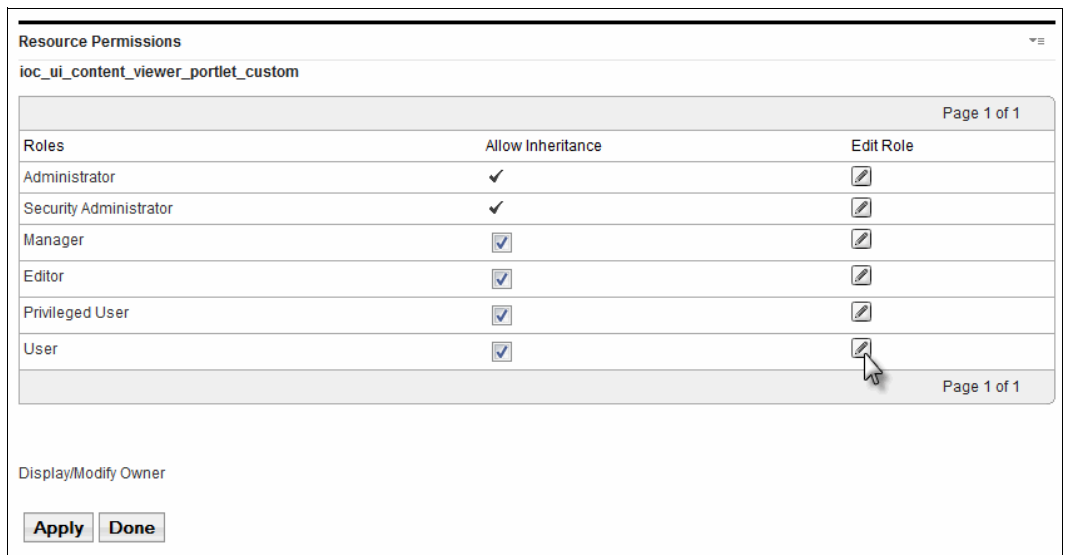


Figure 6-29 Editing the User role

3. Click **Add** (see Figure 6-30).

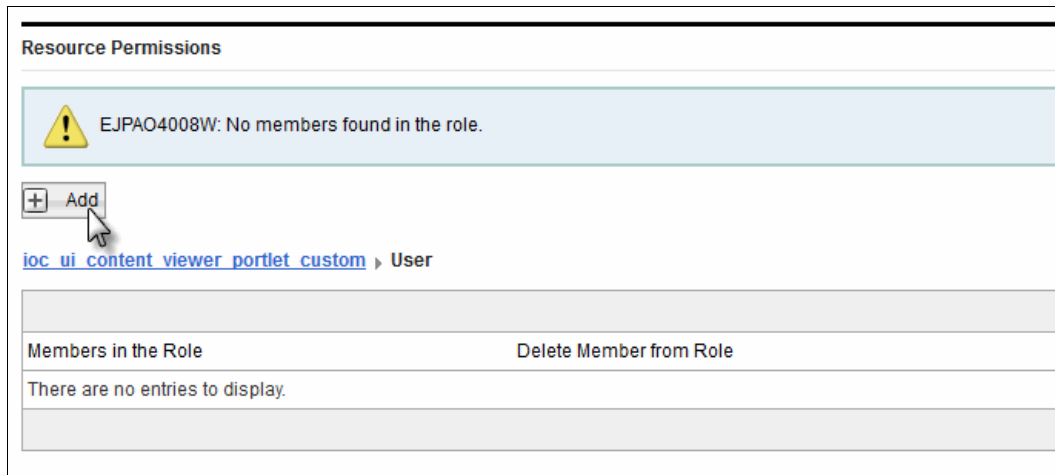


Figure 6-30 Adding users to the User role

4. Select **All Authenticated Portal Users** and click **OK** (see Figure 6-31).

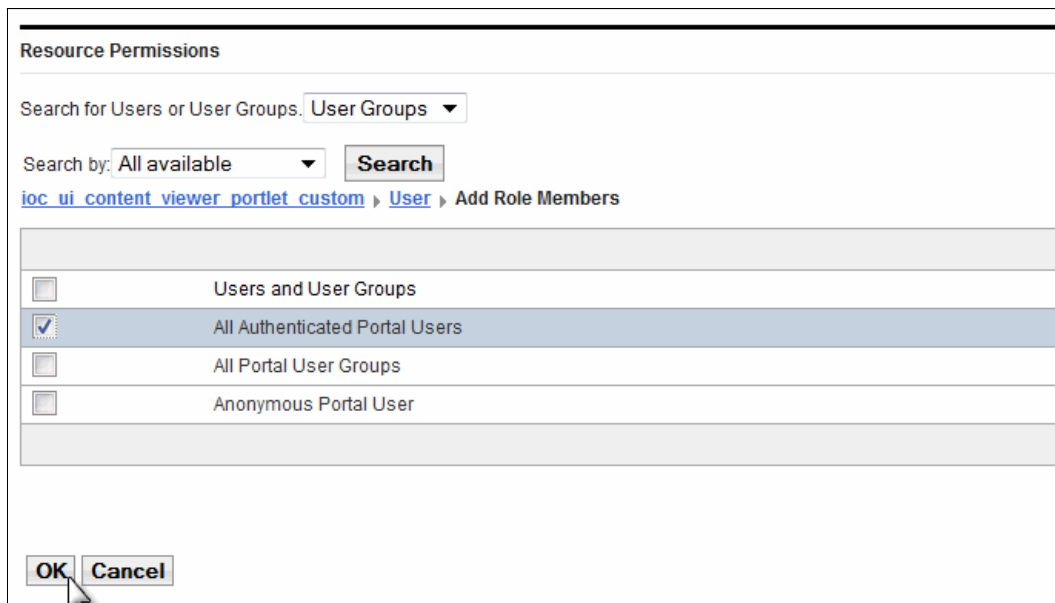


Figure 6-31 Adding All Authenticated Portal Users to the User role

- Go back to the roles configuration by clicking `ioc_ui_content_viewer_portlet_custom` in the path bar (portal breadcrumbs). See Figure 6-32.

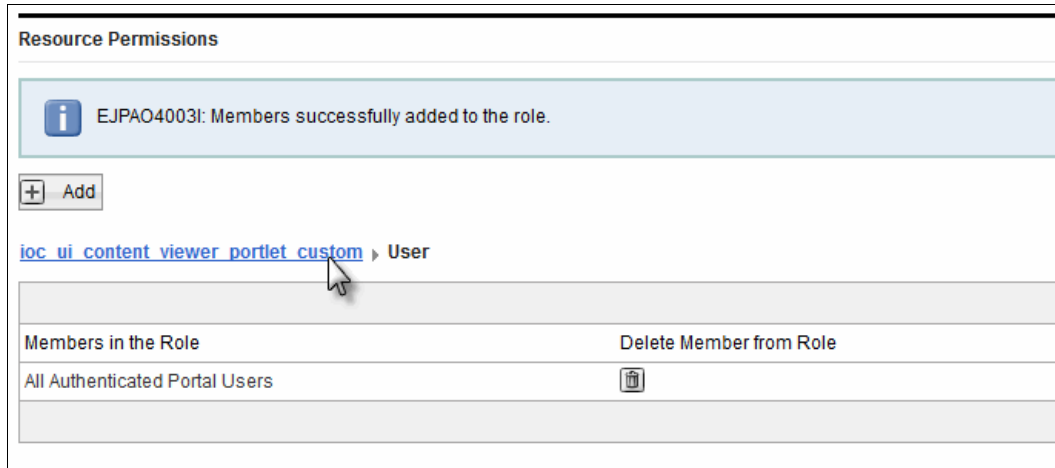


Figure 6-32 Navigating back to user roles

- On the Resource Permissions page, click **Done** to save the changes (see Figure 6-33).

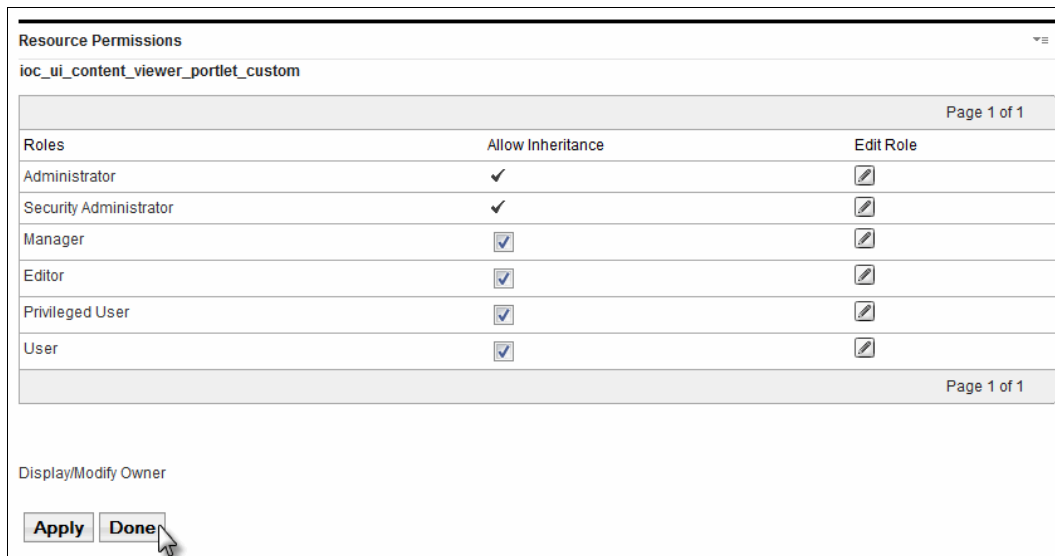


Figure 6-33 Saving the access rights configuration

Adding a new tab with a custom widget into the portlet configuration

Follow these steps:

1. On the Manage Portlets page, locate the new copy of the content viewer portlet (`ioc_ui_content_viewer_portlet_custom` in this example) and click the **Configure portlet** icon (highlighted in Figure 6-34).

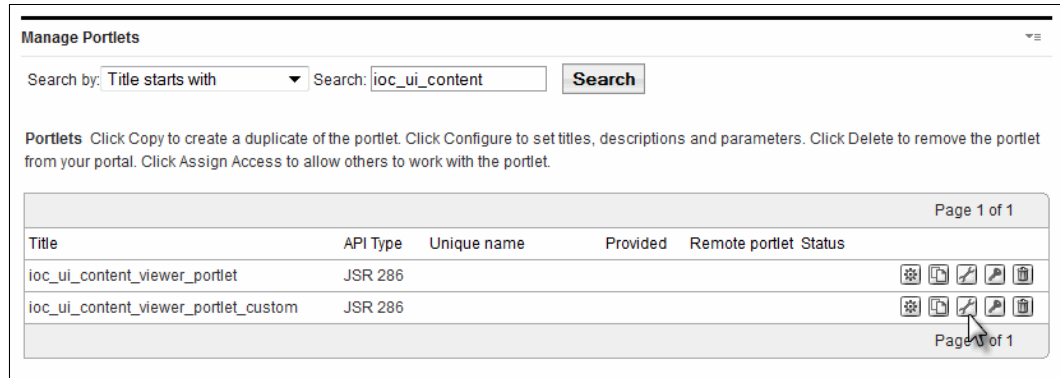


Figure 6-34 Configuring the new copy of the content viewer portlet

2. Enter the configuration parameter:
 - a. In the New Preference field, enter `contentViewTab-ContentViewWidget`.
The `contentViewTab` prefix is a required prefix for the configuration of new tabs. All the characters after the `contentViewTab` prefix form a unique identifier for the tab.
 - b. In the New value field, enter the parameter value shown in Example 6-3.

Example 6-3 Content viewer portlet configuration parameter value example

```
PATH_NAME:example,PATH_LOCATION:/sample/custom/contentviewer/js/,MODULE:example/test/contentviewerWidget,TAB_ORDER:3
```

Note: The parameter property values correspond to the custom application code. Each property is defined:

- ▶ **PATH_NAME:** The name of the path to the new content. It can be used as a unique identifier for the group of widgets within one web application.
- ▶ **PATH_LOCATION:** The absolute path to the new `.js` file with widget code.
- ▶ **MODULE:** The name of the module, which is constructed from the value of the `PATH_NAME` parameter and the value that was used in the `declare` function for the `className` argument. For more details about the `declare` function, see 6.2.3, “Implementation details” on page 198.
- ▶ **TAB_ORDER:** The order in which the tab is displayed in the content panel. The order starts from 0, where 0 is the first tab that is displayed.

For more information, see *Customizing the content panel* at this website:

<http://bit.ly/1fNNvgD>

c. Click **Add** (see Figure 6-35).

Manage Portlets

Configure portlet: `ioc_ui_content_viewer_portlet_custom`
Web module: `ioc_ui_content_viewer_portlet.war`

Preference and Values Enter a new preference and value pair in the blank fields to create a new preference for this portlet, or remove a preference and value. Click OK to keep your changes or Cancel to quit.

New Preference: New value: * Add

Preference	Value
<code>contentViewTab-List</code>	<code>MODULE:ioc/com/ibm/ioc/list/ListView,TAB_ORDER:2</code>
<code>contentViewTab-LocationMap</code>	<code>MODULE:ioc/com/ibm/ioc/locationmap/LocationMap,TAB_ORDER:2</code>
<code>contentViewTab-Map</code>	<code>MODULE:ioc/com/ibm/ioc/map/MapViewWrapper,TAB_ORDER:2</code>

I want to set titles and descriptions.

Enable parallel rendering

Cache Scope for HTTP and fragment caches

Non-shared cache for a single user

Share cache across all users (not applicable if "cache always expires" option is selected below)

Cache Expiration for HTTP and fragment caches

Portlet cache always expires

Portlet cache never expires

Portlet cache expires after this many seconds

OK Cancel

Figure 6-35 Adding a new parameter to the content viewer portlet configuration

3. Click **OK**.

Creating and configuring a new portal page

This step is described in detail in 6.9, "Creating custom solution pages" on page 257. Make sure that the name of a *copy* of the content viewer portlet is used when adding the portlet to the second pane of the new portal page. In this example, the name of the copy of the content viewer portlet is `ioc_ui_content_viewer_portlet_custom`.

After creating a new page and configuring the new portlet, the result will be similar to the custom tab in the custom viewer portlet shown in Figure 6-1 on page 192.

6.3 Creating custom data source actions

Data source actions are a set of user actions that can be applied to a data item (from a data source) and visualized on the user interface (UI), as shown in Figure 6-36. IBM Intelligent Operations Center comes with many data source actions, such as canceling a data item, updating a data item, launching an SOP, and many more.

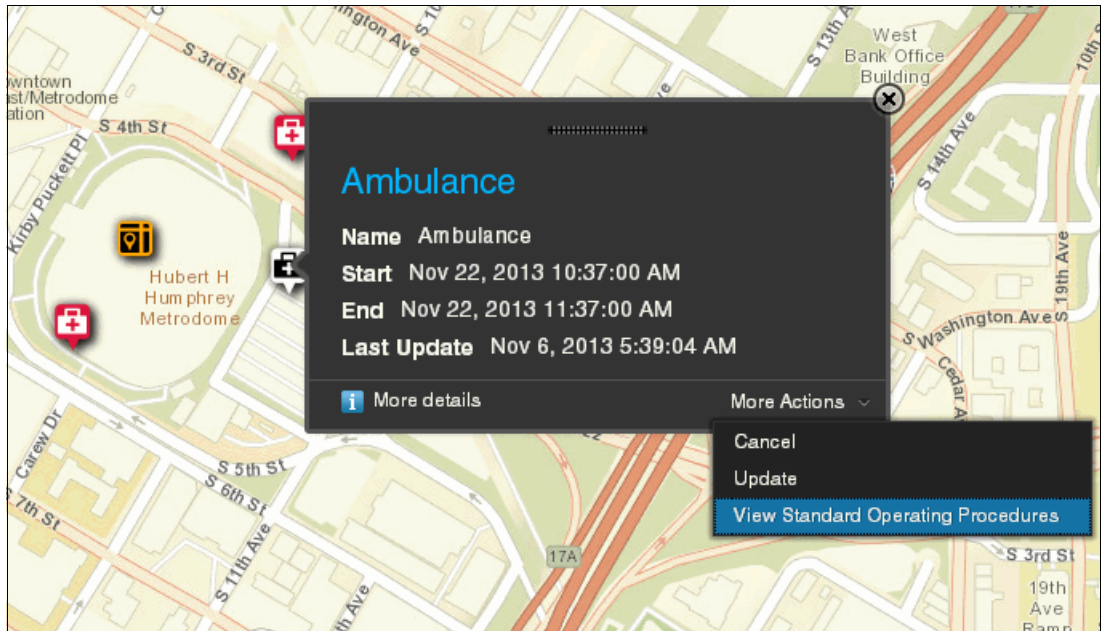


Figure 6-36 Data source actions

In addition to using the predefined actions, custom data source actions can be created to address specific business requirements. The custom actions are integrated in a seamless manner into IBM Intelligent Operations Center. They appear the same as the other predefined actions, in both the Operations view and the data source configuration tool under Administration → Solution Administration → Configuration Tools → Data Sources.

6.3.1 Implementation summary

Custom actions are implemented using JavaScript code as Dojo modules. The following steps summarize the process of adding a custom action to IBM Intelligent Operations Center.

1. Create the custom action Dojo module.
2. Deploy the custom action web module to IBM Intelligent Operations Center.
3. Add a data record to the IBM Intelligent Operations Center database to integrate the custom action.

The sample used in this section is `sample_ioc_custom_actions_web`. See Appendix B, “Samples provided with this book” on page 383.

6.3.2 Setting up the development environment

The following development tools must be set up for this example:

- ▶ IBM Rational Application Developer V8.5. This tool is used to develop and package the custom action Dojo module.
- ▶ DB2 client application: There are multiple tools that can connect to a DB2 database and run queries against it. Developers can use any of the following tools:
 - IBM Data Studio: Provides an integrated environment for database development and administration. IBM Data Studio can be downloaded at no charge from this website:
<http://www-03.ibm.com/software/products/en/data-studio/>
 - DB2 command line processor: This tool can also be used to run queries against the IBM Intelligent Operations Center database.
 - Data Source Explorer view in IBM Rational Application Developer.

The database configuration is simple when adding a custom action. The Data Source Explorer view provides all the necessary features and it is the chosen tool in the example scenario.

6.3.3 Implementation details

This section includes step-by-step instructions for creating custom data source actions.

Creating the custom action Dojo module

The custom action implementation is written as a Dojo module that extends an existing IBM Intelligent Operations Center Dojo widget called `BaseMoreAction`.

Implement a custom action:

1. In IBM Rational Application Developer, create a new Dynamic Web Project by clicking **File** → **New** → **Dynamic Web Project**.
2. Type `CameraFeedAction` in the Project name field and click **Next**.
3. Accept the default values for all other settings and click **Finish**.
4. Create a folder to contain the JavaScript resources:
 - a. In the Project Explorer view, expand the **CameraFeedAction** project.
 - b. Right-click the **WebContent** folder and select **New** → **Folder**. Name the folder `js`.
 - c. Click **Finish**.
5. Under the `js` folder created in step 4, create the package hierarchy that will be used to contain and organize the JavaScript resources. In this example, create a simple hierarchy `/sample/`.

Note: The IBM Intelligent Operations Center actions use the package hierarchy `com/ibm/ioc/moreactions`. This hierarchy can be examined on the IBM Intelligent Operations Center Portal server:

```
/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_theme_ear.ear/ioc_theme_static_js_web.war/js
```

At this point, everything is prepared to create the custom action Dojo module.

6. Create the custom action Dojo module.

The new module extends the existing IBM Intelligent Operations Center module named `ioc/com/ibm/ioc/moreactions/BaseMoreAction`:

- a. Right-click the **js/sample** folder created in step 5 on page 220 and select **New** → **JavaScript Source File**. Name the file `CameraFeed.js`.
- b. Click **Finish**.

7. Open the **CameraFeed.js** file and add the code listed in Example 6-4 to it. This example code performs these functions:

- Declares a new Dojo module named `test.CameraFeed`.
- Extends the `BaseMoreAction` module.
- Implements only a single function named `performAction`. This function is called when the user clicks More Actions and then the action from the data item preview card as shown in Figure 6-39 on page 226.

Two parameters are passed to the `performAction` function when called:

- The first parameter is the data record.
- The second parameter is the preview card instance.

The function in this example displays an empty Dojo dialog.

Example 6-4 Custom action source file

```
define([ "dojo/_base/declare", "ioc/com/ibm/ioc/moreactions/BaseMoreAction",
        "dijit/Dialog", "dijit/form/Button" ], function(declare,
        BaseMoreAction, Dialog, Button) {

    return declare("test.CameraFeed", [ BaseMoreAction ], {
        constructor : function(args) {
            declare.safeMixin(this, args);
        },

        performAction : function(dataRecord, previewCard) {
            previewCard.hide();
            var title = dataRecord.Name + " " + "video feed";
            var cDialog = new Dialog({
                title : title,
                style : "width: 300px"
            });
            var cButton = new Button({
                label : "Close",
                onClick : function() {
                    cDialog.destroy();
                }
            });
            cDialog.addChild(cButton);
            cDialog.show();
        }
    });
});
```

The code for this widget can be retrieved from `sample_ioc_custom_actions_web` or the technote *Sample Dojo script for Intelligent Operations Center V1.6 data source custom action* at <http://www-01.ibm.com/support/docview.wss?uid=swg21653065>.

Deploying the custom action to IBM Intelligent Operations Center

After the development of the custom action is complete, the next step is to deploy it to IBM Intelligent Operations Center.

The web module containing the action is deployed to the WebSphere Portal server running in the IBM Intelligent Operations Center application server so that it is accessible from the IBM Intelligent Operations Center runtime environment.

Deploy the web module. This process is described in more detail in “Installing the new application in IBM WebSphere Application Server” on page 206.

1. Set the web module context root:
 - a. Right-click the **CameraFeedAction** project and select **Properties** on the properties page.
 - b. Go to Web Project Settings and type the context root path. In this example, use the context root path `sample/custom/more/actions`.
 - c. Click **OK** to close the dialog.
2. Export the web module web archive (WAR) file:
 - a. Right-click the **CameraFeedAction** project and select **Export** → **WAR file**.
 - b. Type the WAR file name in the Destination field, for example, `CameraFeedAction`.
 - c. Click **OK**.
3. Deploy the WAR file to the application server.

Open the WebSphere Application Server administration console. From the IBM Intelligent Operations Center portal interface, navigate to **Solution Administration** → **System administration** → **Administrative consoles** → **Application server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Sametime Proxy).

Note: This is the application server console for WebSphere Application Server where the Portal server is installed. Use the administrator username and password to log in.

4. Click **Applications** → **New Application** → **New Enterprise Application**.
5. In the next window (Preparing for the application installation- Path to the new application), select **Local file system**. Click **Choose File** to select the WAR file exported in step 2, for example, **CameraFeedAction.war**. Click **Next**.
6. In the next window (Preparing for the application installation - How do you want to install the application?), select **Fast Path** and click **Next**.
7. Map the modules to servers.

There are many configuration steps involved in installing an application on WebSphere Application Server.

Skip to Step 2: Map modules to servers and make sure that your application is mapped to both the **Portal Cluster** and **ihserver1**, as shown in Figure 6-37.

Important: The custom action will not work without the correct server mappings.

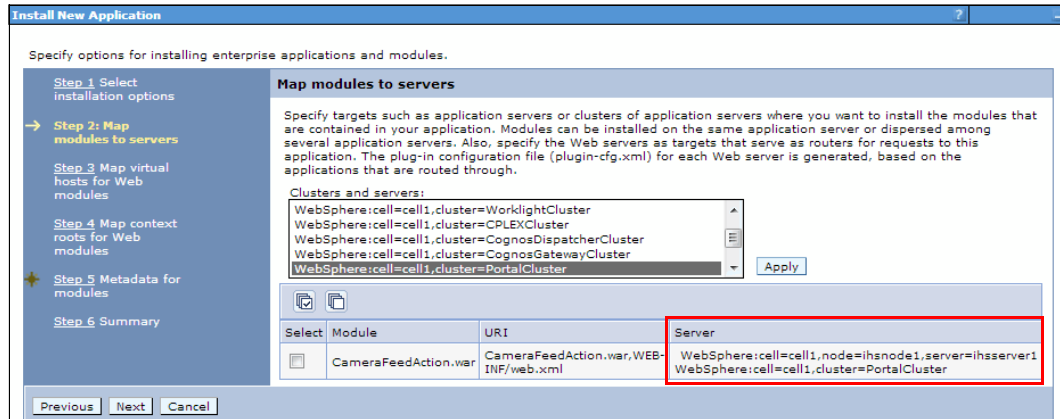


Figure 6-37 Server mappings

8. Map context roots for web modules.

Skip to Step 4: Map context roots for web modules and make sure that the context root is set correctly. Click **Next** until the deployment is complete, and then, click **Save directly to the master configuration**.

9. Select the **CameraFeedAction_war** application from the list of applications and then click **Start**.

10. Test the deployment.

Access the custom action Dojo module from the browser. The URL to the JavaScript file uses the following format:

```
https://<ioc_ibs>/<WebModuleContextRoot>/<FolderStructureUnderWebContent>/CameraFeed.js
```

This example uses this URL:

```
https://<ioc_ibs>/sample/custom/more/actions/js/test/CameraFeed.js
```

If the content of the JavaScript file loads in the browser correctly, the deployment is successful.

Adding the custom action data record

The custom action web module is now deployed and started on the IBM Intelligent Operations Center server.

The next step is to configure IBM Intelligent Operations Center to access the custom action and to display it. This configuration consists of adding data records to two tables in the IBM Intelligent Operations Center database:

- ▶ Add a data record to table IOC.Available_Extensions with the SQL statement shown in Example 6-5.

Example 6-5 Adding custom action record to the available extensions table

```
INSERT INTO IOC.Available_Extensions
(Name, Type, Disabled, Multiple, PathLocation, PathName, Module)
VALUES
('CameraFeed', 'ACTION', '1', '0', '/sample/custom/more/actions/js', 'exmp',
'exmp/test/CameraFeed');
```

Table 6-2 explains the parameters used in the SQL statement.

Table 6-2 Adding custom action parameters

Name	Description	Sample value
Name	Name of the custom action.	CameraFeed
Type	Type of the custom action. It can be an action of a preview card add-on.	Action
Disabled	If set to 1, by default, this action will be disabled in the data source tool. (The Use? property corresponding to this action will be set to No on the Actions tab.)	1
Multiple	If set to 1, this action can be executed on multiple data records at one time. If Multiple is set to 1, the performAction function receives an array of data records as a parameter instead of a single item. This action will also appear in the More Actions menu on the upper-right corner of the page (see 6.8, “Customizing the More Actions menu” on page 255).	0
PathLocation	Path to the module location. This is the portion of the path that is given the alias name specified in PathName.	/sample/custom/more/actions/js
PathName	An alias name given to the PathLocation. Module path (the next parameter) will use this alias.	exmp
Module	Path of the custom action module relative to the registered PathLocation.	exmp/test/CameraFeed.js

- ▶ Add custom action name translations. At a minimum, add the *default* locale. Use the insert statement shown in Example 6-6.

Example 6-6 Adding action name translations

```
INSERT INTO IOC.I18N_RESOURCE (GROUP, LOCALE, KEY, VALUE) VALUES
('PreviewCard', 'default','action_CameraFeed', 'Camera Feed');
```

The key parameter must start with the string `action_` and then add the action name from the `AVAILABLE_EXTENSIONS` table to it. In this example, it is `action_CameraFeed`.

6.3.4 Visualizing the results

The newly created custom action will only be available for data sources created after the custom action was created. Create a new data source as described in Chapter 3, “Integrating with data sources” on page 45.

During the new data source creation, the new action can be configured on the Actions tab of the data source configuration tool. Figure 6-38 shows that you can select whether the CameraFeed custom action needs to be displayed for the data source.

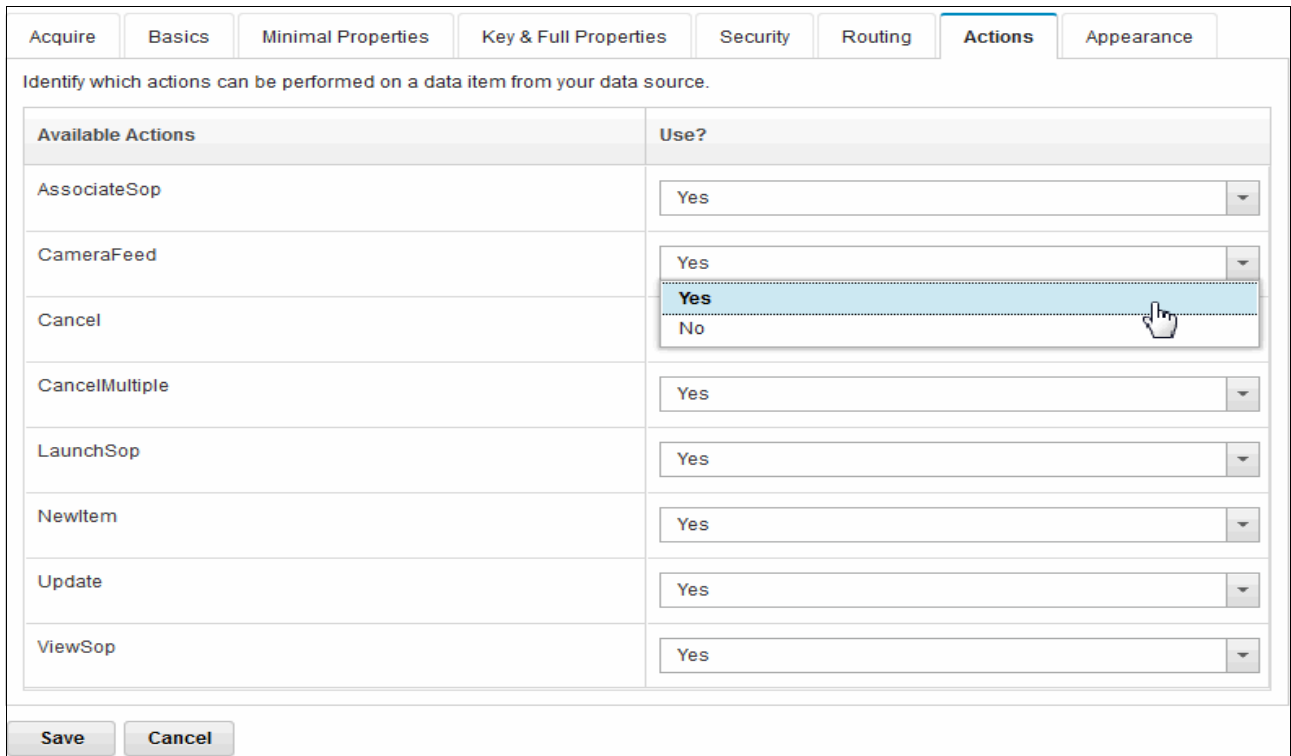


Figure 6-38 Custom actions in data source configuration tool

If the action is configured to be used, it will be displayed in the More Actions menu as shown in Figure 6-39 on page 226.

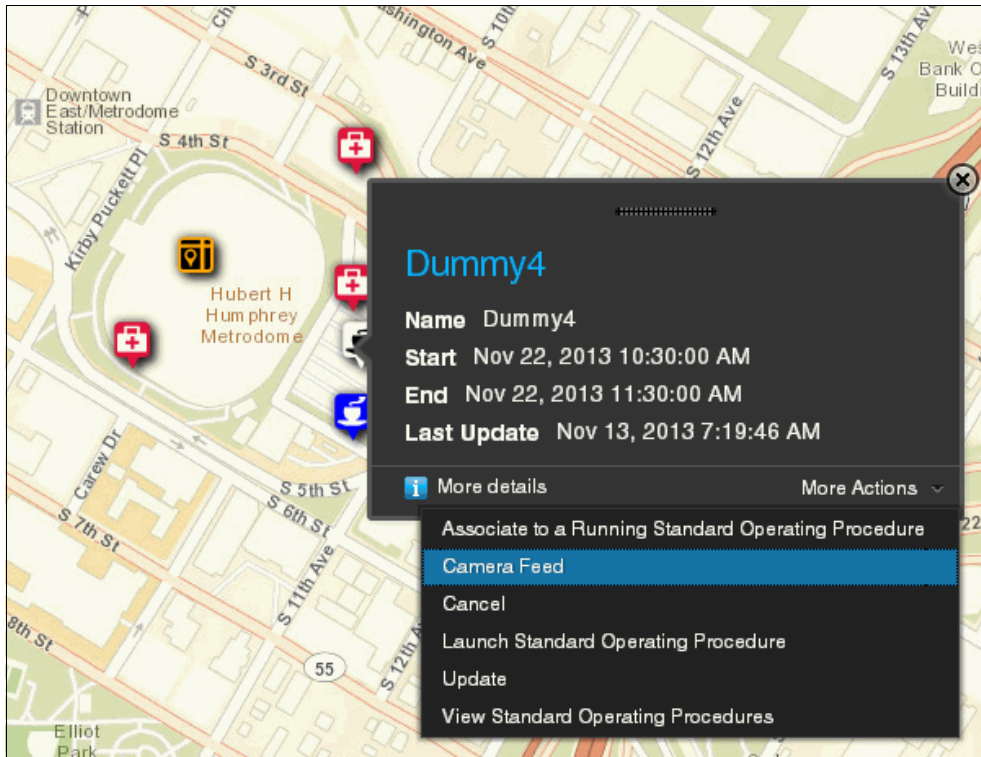


Figure 6-39 Custom action displayed in the list of actions in the More Actions menu

Clicking the Camera Feed action opens the empty dialog created in this example as shown in Figure 6-40.

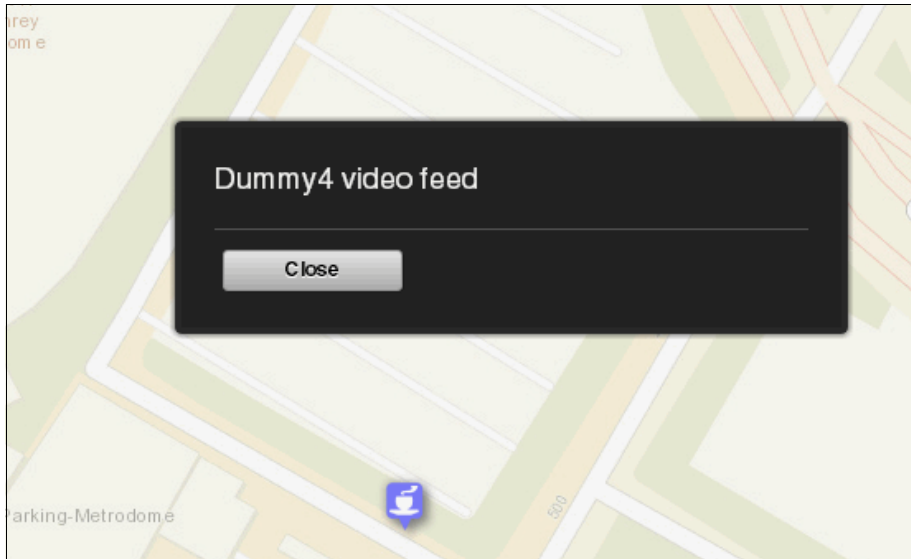


Figure 6-40 Displayed result of custom action

6.4 Adding filter-driven result charts

The default filter-driven result charts are implemented using Dojo chart widget technology. There are two Dojo chart widgets available by default:

- ▶ Total results by data source: This chart shows event count per data source.
- ▶ Data source results by time: This chart visualizes event count per data source per time period (by month, by week, or by day).

These Dojo chart widgets are not meant as a replacement for the Cognos reports. They provide another way to display common information and trends across all currently selected data sources using data that spans all of them. For information about creating Cognos charts for IBM Intelligent Operations Center, see Chapter 10, “Creating and viewing reports” on page 349.

You can create a new custom Dojo chart widget by extending the standard `ioc/com/ibm/ioc/contentviewer/View` widget.

Six functions can be redefined; these functions are called when users act on the data source. The functions listed in Table 6-1 on page 200 can be used to change the displayed data and trigger the chart redraw process.

Custom code has access to the variables listed in Table 6-3.

Table 6-3 Variables that are passed to the standard `ioc/com/ibm/ioc/contentviewer/View` widget

Variable	Description
<code>datasourceStore</code>	This variable is passed to the <code>addDatasource</code> and <code>updateDatasourceCriteria</code> functions. It contains a collection of data items for each data source.
<code>datasourceConfig</code>	This variable contains data source property information.

Example 6-7 illustrates how to redefine the function `addDatasource(datasourceConfig, datasourceStore)` and use the variables `datasourceStore` and `datasourceConfig` to get the data source labels and total number of data sources.

Example 6-7 Redefining “`addDatasource`” function to redraw the custom chart widget

```
// [...]
addDatasource: function(datasourceConfig, datasourceStore){
    var methodName = "addDataSource";
    this._logger.traceEntry(methodName);

    // Get new data source label info
    var labelObj = JSON.parse(datasourceConfig.label);
    this._checkedDataSourceStore.put({id: datasourceConfig.id, name:
i18nRes_DataSource[labelObj.key]});

    // Add new data source data into “_loadedDatasourcesCache”
    when(datasourceStore.ready, lang.hitch(this, function() {
        this._loadedDatasourcesCache[datasourceConfig.id] = {
            store: datasourceStore,
            config: datasourceConfig
        });
    });
};
```

```

    // Get total number of data sources into "total"
    var total = 0;
    array.forEach(this._checkedDatasourceStore.query(), lang.hitch(this,
function(item) {
    total += this._loadedDatasourcesCache[item.id].store.count.value;
    });

},
// [...]
```

Note: Find more examples in the source code of the charts included with IBM Intelligent Operations Center. The examples are available on the application server at this directory:

```

/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_portal_ear.ear/ioc_ui_cha
rting_portlet.war/_ioc_ui_charting_portlet/js/ChartView.js
```

Because the charts are filter-driven, adding new charts requires you to check for updates in the data sources or filter changes and to redraw the charts if new data is provided.

There are three events that also need to be monitored in the code:

- ▶ Add to the store and delete from the store events: These events are handled by placing a watch function on the `datasourceStore.count` variable, as shown in Example 6-8.
- ▶ Update to the store events: These events are handled by a topic subscription to the `/ibm/ioc/reststore/update` topic, as shown in Example 6-9.

Example 6-8 Watching for update and delete events

```

// [...]
array.forEach(this._checkedDatasourceStore.query(), lang.hitch(this,
function(item) {
// Init new count watcher
    if(!this._loadedDatasourcesCache[item.id].watchHandler) {
        this._loadedDatasourcesCache[item.id].watchHandler =
            this._loadedDatasourcesCache[item.id].store.count.watch("value",
// Calling function to redraw the chart in case of changes
                this._generateChart("custom"));
    }
}));
// [...]
```

Example 6-9 Subscribing to /ibm/ioc/reststore/update and listening for updates

```

// [...]
// Listen to RestStore updates by subscribing to the Dojo topic and call
// _storeUpdated() function to redraw the chart

topic.subscribe("/ibm/ioc/reststore/update", lang.hitch(this,
this._storeUpdated));
// [...]
```

The default Dojo chart widgets (the ones included with IBM Intelligent Operations Center) show the statistics for all currently filtered data items, which means that they can only use data that spans all data sources.

The default Dojo chart widgets built on multiple data sources can only use the properties that are common to all the data sources, that is, the minimal properties: name, location, start date, end date, and a few other optional properties. Custom Dojo chart widgets can use other properties, but in this case, the developer must ensure that the same properties are present in all the data sources.

By rewriting the functions described in Table 6-1 on page 200, chart generation using the minimal properties becomes relatively simple:

1. Use the appropriate store information passed through the `datasourceStore` variable to generate the data that is displayed in the charts. You can use the same approach to generate the labels for the x/y-axis if necessary.
2. Create the standard Dojo chart using the `dojox.charting` library. For information, see the *Dojo documentation* at this website:

<http://dojotoolkit.org/reference-guide/1.8/dojox/charting.html#dojox-charting>

Both default Dojo chart widgets (included in the IBM Intelligent Operations Center UI) are part of the same `ioc_ui_charting_portlet` portlet. New Dojo chart widgets must be added to the same portlet and then displayed under the default chart widgets.

The Dojo chart widget uses a `<div>` tag to display the chart in HTML. This tag can be added to the `ChartingPortletView.jsp` file located on the application server in the following directory:

```
/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_portal_ear.ear/ioc_ui_charting_portlet.war/_ioc_ui_charting_portlet/jsp/html/.
```

Because all default Dojo chart widgets use the same data, the legend remains the same for all of them. Custom Dojo chart widgets can use this legend as well, in case they visualize the same data as the default charts.

6.4.1 Custom filter-driven result chart creation example

This section describes an example of creating a custom filter-driven result chart using Dojo chart widget technology.

Implementation summary

The following steps summarize the process for creating a custom filter-driven result chart:

1. Create a custom Dojo chart widget that implements the filter-driven result chart.
2. Customize the `ChartingPortletView.jsp` to add a place holder `<div>` tag and link to the file with new chart widget.

Setting up the development environment

Any text editor that supports UTF-8 encoding can be used to create the Dojo widget and edit the `ChartingPortletView.jsp` file on the application server file system.

Implementation details

Note: The full version of the example source code for `CustomChart.js` and `ChartingPortletView.jsp` is available in the file `sample_ioc_custom_charts` with the examples provided with this book. See Appendix B, “Samples provided with this book” on page 383.

Creating a custom Dojo chart widget

Example 6-10 provides sample code for the custom Dojo chart widget. Notice the following points in Example 6-10:

- ▶ Define the chart object by passing the `div` object on which the chart will be displayed.
- ▶ The `addPlot` function is used to define the appearance of the chart, specifying, for example, the chart type, the radius for the pie chart, the color, labels, and more.

Follow the steps to create a new Dojo chart widget:

1. Create the `js` directory under
`/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_portal_ear.ear/ioc_ui_charting_portlet.war/_ioc_ui_charting_portlet.`
2. Create `CustomChart.js` in the `js` directory created in the previous step. The full path is
`/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_portal_ear.ear/ioc_ui_charting_portlet.war/_ioc_ui_charting_portlet/js/CustomChart.js.`
3. Add the code from Example 6-10 to the `CustomChart.js` created in the previous step.

Example 6-10 Defining the chart and its appearance

```
/**
 * Defining the required widgets
 */
require(["dojo/dom",
        "dojo/dom-construct",
        "dojo/dom-class",
        "dojox/charting/Chart",
        "ioc/com/ibm/ioc/charting/themes/ReportChartingTheme",
        "dojox/charting/plot2d/Pie"
        ],
        function(dom, construct, domClass, Chart, ReportChartingTheme){
/**
 * Creating a div for the chart inside another div with
 * id=ioc_custom_chart_container
 */
var dchartDiv = construct.create("div", null,
dom.byId("ioc_custom_chart_container"), "last");
/**
 * Adding CSS style class
 */
        domClass.add(dchartDiv, "ioc_chart_display");
/**
 * Creating an object for chart using div defined earlier
 */
        this._customChart = new Chart(dchartDiv);
        // Set the theme
        this._customChart.setTheme(ReportChartingTheme);
```



```

/**
 * Adding the default plot
 */
    this._customChart.addPlot("default", {
        type : "Pie",
        radius : 100,
        fontColor : "white",
        labelOffset : -20
    });
/**
 * Adding the series of sample data
 */
    var chartData = [{
        x : 1,
        y : 19021
    }, {
        x : 1,
        y : 12837
    }, {
        x : 1,
        y : 16122
    }
    ];
    this._customChart.addSeries("Series Name", chartData);
/**
 * Rendering the chart!
 */
    this._customChart.render();
});

```

Customizing the ChartingPortletView.jsp file

The ChartingPortletView.jsp is the main file that defines how to render the Dojo chart widgets inside the charting portlet. Example 6-11 on page 232 contains a series of <div> tags that are placeholders for the default chart widgets.

Note: The full version of the example source code for CustomChart.js and ChartingPortletView.jsp is available in the file sample_ioc_custom_charts with the examples provided with this book. See Appendix B, “Samples provided with this book” on page 383.

Follow these steps to customize the `ChartingPortletView.jsp` file:

1. To add a new widget, add a new `<div>` as shown in Example 6-11.

Example 6-11 Adding placeholder `<div>` tag to `ChartingPortletView.jsp`

```
// [...]
<div id="ioc_chart_analytics" class="ioc_chart_analytics" tabindex="0">
  <div id="ioc_chart_container" class="ioc_chart_container ioc_chart_hidden"
role="region">
  // [...]
  <div id="ioc_custom_chart_container" class="ioc_chart_inner_container">
    <div class="ioc_chart_title">Demo Chart</div>
  </div>
  </div>
  // [...]
</div>
// [...]
```

2. Add a link to the `CustomChart.js` file, which is created in step 2 on page 230, immediately after the last `</script>` tag and before the first `<div>` tag. See Example 6-12.

Example 6-12 Adding a link to `CustomChart.js` file

```
// [...]
</script>

<script
src="<%=portletContext%>/_ioc_ui_charting_portlet/js/CustomChart.js"></script>

<div id="ioc_chart_analytics" class="ioc_chart_analytics" tabindex="0">
// [...]
```

3. To test the custom Dojo chart widget, access the Operator view and click the **Chart** tab on the right side of the panel. See Figure 6-41.

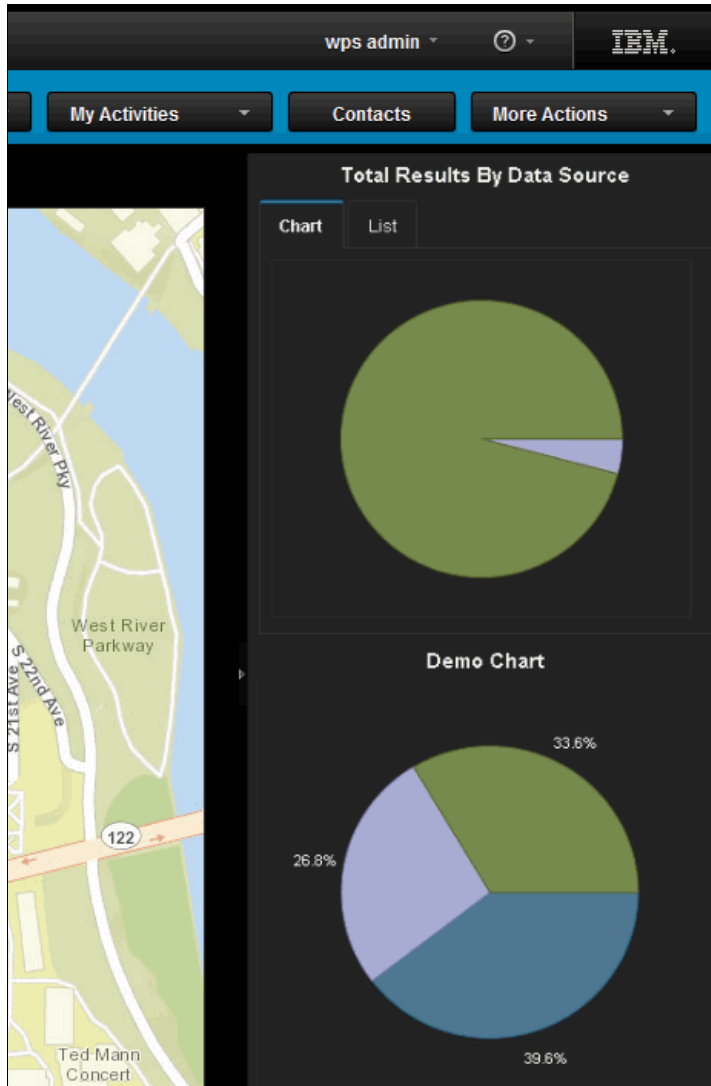


Figure 6-41 Testing the custom Dojo chart widget on the right side of the panel

6.5 Customizing preview cards

The *preview cards* for data items show properties that were defined as key properties when the data source was created. By default, all the properties are in text format; however, it is possible to add other types of information, such as images, links, and video.

The following samples are used in this section:

- ▶ test_ioc_ui_test_custom_preview_card_portlet
- ▶ test_ioc_ui_test_custom_preview_card_portlet_db2
- ▶ test_ioc_ui_test_custom_preview_card_portlet_ear

For the full list of samples included with this book, see Appendix B, “Samples provided with this book” on page 383.

6.5.1 Implementation summary

Custom preview cards are implemented using JavaScript code as Dojo modules. The following steps summarize the process of adding a custom preview card to IBM Intelligent Operations Center:

1. Create the custom web application for the preview card.
2. Deploy the custom preview card web application to WebSphere Application Server.
3. Create a data source with properties that contain URLs for images, videos, or other custom data that need to be displayed in the custom preview card.
4. Add a data record to the IBM Intelligent Operations Center database to integrate the new data source with the custom preview card.

6.5.2 Setting up the development environment

The following development tools must be set up for this example:

- ▶ IBM Rational Application Developer V8.5. This tool is used to develop and package the custom action Dojo module.
- ▶ DB2 client application: There are multiple tools that can connect to a DB2 database and run queries against it. Developers can use any of the following tools:
 - IBM Data Studio: Provides an integrated environment for database development and administration. IBM Data Studio can be downloaded at no charge from this website: <http://www-03.ibm.com/software/products/en/data-studio/>
 - DB2 command line processor: This tool can also be used to run queries to the IBM Intelligent Operations Center database.
 - Data Source Explorer view in IBM Rational Application Developer.

The DB2 command line processor is used in this example.

6.5.3 Implementation details

The implementation details are described.

Creating a custom web application for the preview card

The standard user interface of IBM Intelligent Operations Center is built using the Dojo framework and it can be extended by using standard Dojo extension capabilities, such as widgets.

To create a custom widget that will represent a custom preview card, extend the default widget called `ioc/com/ibm/ioc/previewcard/_PropertiesPane` by using the `declare` function from the `dojo/_base/declare` module.

Note: The `ioc/com/ibm/ioc/previewcard/_PropertiesPane` default widget is defined in the IBM Intelligent Operations Center portal theme file:

```
/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_theme_ear.ear/ioc_theme_s  
tatic_js_web.war/js/com/ibm/ioc/previewcard/_PropertiesPane.js
```

Table 6-4 describes the properties in the sample data source that the preview card script uses to display an image, show an external page in an iframe HTML tag, or show the headlight data item property value as a web link. These properties contain the URL to the data to be displayed in the preview card.

Table 6-4 List of properties implemented in example

Property	Description
IMAGEFILE	URL to the image file that is displayed in the preview card in the tag. See example in Figure 6-4 on page 195.
IFRAME	URL to the web page that is shown in the preview card inside the <iframe> HTML tag.
YOUTUBE	URL to the video on youtube.com.
FLVIDEO	URL to the video file that is played using the dojox/av/FLVideo library. For more information, see the Dojo documentation at this website: http://dojotoolkit.org/reference-guide/1.9/dojox/av/FLVideo.html
LINK	URL used as link (HTML <a> tag).

To create a new custom property, add a property label to the `_createNode` function and implement the handler for this property to define how it will be represented on the preview card.

For more information, see the comments in the source code in the `test_ioc_ui_test_custom_preview_card_portlet\WebContent\js\com\ibm\ioc\previewcard_CustomSamplePropertiesPane.js` file.

Creating and deploying the custom preview card application

The process to create and deploy the custom preview card is explained.

Note: The steps in this section are the same as the steps in 6.2.3, “Implementation details” on page 198 and “Generating and deploying the EAR file” on page 201. For the corresponding steps in this section, see the figures in those sections.

Importing the sample code into Rational Application Development workspaces

Follow these steps:

1. Start Rational Application Developer and click **File** → **Import**.
2. Click **General** → **Existing Projects into Workspace** and then click **Next**.
3. Click **Select root directory** and click **Browse**:
 - a. Navigate to the folder with the samples: `/SAMPLES/8201PortalVis`.
 - b. Select `test_ioc_ui_test_custom_preview_card_portlet` and `test_ioc_ui_test_custom_preview_card_portlet_ear`.
 - c. Click **Finish**.

Exporting the EAR file

Follow these steps:

1. In Project Explorer, right-click **test_ioc_ui_test_custom_preview_card_portlet_ear** and select **Export** → **EAR file**.
2. Click **Browse** next to the Destination field and select the path to the folder where the exported file needs to be placed. Click **Finish**.

Installing the new application in IBM WebSphere Application Server

Follow these steps:

1. Log in to the WebSphere Application Server 8.0 Deployment Manager administration console on the application server with the administrative account. You can access the console from the IBM Intelligent Operations Center portal:

Click **Administration** → **Solution Administration** → **System Administration Administration** → **Consoles** → **Application server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Sametime Proxy).

Alternatively, you can access the console with the `https://<ioc_app>:9044/ibm/console/` URL.
2. Click **Applications** → **New Application**.
3. Click **New Enterprise Application**.
4. For Path to the new application, select **Local file system** and click **Choose File**. Navigate to the EAR file exported in “Exporting the EAR file” on page 236, and click **Next**.
5. Select **Fast Path** and click **Next**.
6. Accept the defaults for all fields for Step 1: Select installation options, and click **Next**.
7. Follow these steps in Step 2: Map modules to servers:
 - a. Select the **test_ioc_ui_test_custom_preview_card_portlet.war** module.
 - b. From the list under Clusters and servers, select this cluster and this server:
 - **WebSphere:cell=cell1,node=ihsnode1,server=ihsserver1**
 - **WebSphere:cell=cell1,cluster=PortalCluster**
 - c. Click **Apply**, and then, click **Next**.

The web module inside the Java Platform, Enterprise Edition application is deployed to the WebSphere Portal Server. It is accessible through the HTTP server.
8. Accept the default values for Step 3: Metadata for modules, and click **Next**.
9. For Step 4: Summary, click **Finish**.
10. Click **Save**.
11. Click **Save directly to the master configuration**.

Starting and testing the new application

Follow these steps:

1. Navigate to **Applications** → **Application types** → **WebSphere enterprise applications**.
2. Search for and select **test_ioc_ui_test_custom_preview_card_portlet_ear** by clicking the check box next to the application name, and click **Start**.
3. Test the application.

Open a new tab in the browser and enter the URL:

```
https://<ioc_ibs>/ibm/ioc/test/custom/js/js/com/ibm/ioc/previewcard/_CustomSamplePropertiesPane.js
```

The content of the JavaScript file is displayed in the browser.

Configuring custom preview cards in IBM Intelligent Operations Center

The custom preview card application must be associated with the data source for which the preview card displays the key properties of the data items. The example described in this section uses a database data source; therefore, a staging table is created (just for completeness).

After the data source is created, it is associated with the sample preview card.

The following steps summarize the tasks to implement this example:

1. Create a new table in the staging database for the sample data source.
2. Create a new sample data source for the custom preview cards.
3. Associate the custom preview card with the new data source (by making modifications to the `IOC.DataSource_Extensions` table).
4. Test the new preview card in the Operations view.

Creating a new table in the staging database for the sample data source

Follow these steps:

1. Copy scripts from the `test_ioc_ui_test_custom_preview_card_portlet_db2` directory provided in the samples archive with this book to the database server IBM Intelligent Operations Center data server, for example, to `/tmp`.
2. Log in to the data server as `db2inst2`.
3. Create a table for the sample data source by running the following command in the data server command-line interface from the directory `test_ioc_ui_test_custom_preview_card_portlet_db2`:

```
db2 -tvf SAMPLE_CUSTOM_CARDS.dd1
```
4. Populate the new table with sample data by running the following command from the directory `test_ioc_ui_test_custom_preview_card_portlet_db2`:

```
db2 -tvsvf SAMPLE_CUSTOM_CARDS.sql
```

Note: The IBM Intelligent Operations Center data server can be used for testing or development purposes, but you need a separate database instance for the staging database. However, the preferred approach is to separate the IBM Intelligent Operations Center development and test environments from the production environment.

Creating a new sample data source for the custom preview cards

Follow these steps:

1. Log in to IBM Intelligent Operations Center as the administrator and navigate to the data source configuration tool by clicking **Administration** → **Solution Administration** → **Configuration Tools** → **Data Sources**.
2. Create a new data source:
 - a. In the data source configuration tool step 1: Acquire, connect the data source to the database table created in “Creating a new table in the staging database for the sample data source” on page 237.
 - b. In the data source configuration tool step 2: Basics, make note of the value that you enter for the data source unique text identifier property.
 - c. In the data source configuration tool step 4: Key & Full Properties, assign key property status to the IMAGEFILE. (The LINK property can be selected as Key, as well.)
3. Add the new data source to the filter panel so that you can see it on the map.

Note: For details about creating a data source and displaying it on the map, see 3.2.3, “Database table data source” on page 54.

Associating the custom preview card with the new data source

Follow these steps:

1. Locate the new data source in the IOC.DataSource_Extensions table in the data server:
 - a. On the data server, start the **db2** command-line tool and log in as db2inst2.
 - b. Locate the data source created in “Creating a new sample data source for the custom preview cards” on page 238. The data source is in the IOC.DataSource_Extensions table, and it has the *data source unique text identifier* specified in step 2.
 - c. Run an SQL query to find the data source ID by running the SELECT statement shown in Example 6-13, where CCARDS is the data source unique text identifier and 88 is the data source ID returned by the request.

Example 6-13 Finding the data source ID using the data source unique text identifier

```
# su - db2inst2
$ db2 SELECT DATASOURCEID FROM IOC.DataSource_Extensions WHERE ASSOCIATED
LIKE \'CCARDS\'

DATASOURCEID
-----
           88

1 record(s) selected.
```

2. Modify the INSERT statement that is contained in the INSERT_STATEMENT_TO_ACTIVATE_CUSTOM_CARDS.sql script, which is located in the samples test_ioc_ui_test_custom_preview_card_portlet_db2 directory, to include the new data source ID as shown in Example 6-14.

Example 6-14 Insert statement to activate custom preview card for data source ID equal to 88

```
insert into IOC.DataSource_Extensions (DataSourceID, Name, Type, Disabled,
Multiple, PathLocation, PathName, Module) values (88,
'_CustomSamplePropertiesPane', 'PREVIEW_CONTENT_TYPE', '1', '0',
'/ibm/ioc/test/custom/js/js', 'ioc',
'ioc/com/ibm/ioc/previewcard/_CustomSamplePropertiesPane');
```

3. On the data server, start the db2 command-line tool and log in as db2inst2. Add the custom preview card into the IOC.DataSource_Extensions table by executing the modified INSERT_STATEMENT_TO_ACTIVATE_CUSTOM_CARDS.sql script, as shown in Example 6-15.

Example 6-15 Executing INSERT_STATEMENT_TO_ACTIVATE_CUSTOM_CARDS.sql script

```
# su - db2inst2
$ db2 -tvvf INSERT_STATEMENT_TO_ACTIVATE_CUSTOM_CARDS.sql
```

Testing the new preview card on the Operations view

Follow these steps:

1. Log in to IBM Intelligent Operations Center as the administrator and navigate to the Operations view, for example, click **Citywide** → **Supervisor: Operations**.
2. Select the data source with the sample data. Click one of the data items displayed on the map to see the custom preview card, as shown in Figure 6-4 on page 195.

Note: If the data items are not displayed, follow the instructions for troubleshooting in 3.5, “Troubleshooting data sources” on page 64.

6.6 Creating custom portlets

Developers can create custom portlets, deploy them in the IBM Intelligent Operations Center portal server, and seamlessly integrate them with existing IBM Intelligent Operations Center components. The custom portlets can use the predefined standard data feeds that are available with IBM Intelligent Operations Center.

This section describes how to create, configure, and deploy custom portlets for IBM Intelligent Operations Center.

6.6.1 Custom portlet development environment

Custom portlets to be used with IBM Intelligent Operations Center can be developed in IBM Rational Application Developer in the same manner as any other IBM WebSphere Portal portlet.

The IBM Intelligent Operations Center development libraries are needed in the development environment to compile the custom portlet code. The development libraries are available as JAR files on the IBM Intelligent Operations Center application server installation directory under the lib directory, by default, /opt/IBM/ioc/lib.

Prepare the development environment for custom portlet development:

1. Download the IBM Intelligent Operations Center JAR files from the `lib` directory in the application server (by default, `/opt/IBM/ioc/lib`) to the development workstation.
2. Make the JAR files available in the development environment as a shared library.

Follow these steps:

- a. Start IBM Rational Application Developer.
 - b. Click **Window** → **Preferences**.
 - c. From the left pane, expand **Java** → **Build Path**.
 - d. Click **User Libraries** (see Figure 6-42).
3. Click **New** to add a new user library, name it `IOC_JARS`, and click **OK**.
 4. Select the `IOC_JARS` library and click **Add JARS**.
 5. Browse to the JARs downloaded from IBM Intelligent Operations Center server in step 1 and click **Open**. The window looks like Figure 6-42. Click **OK**.

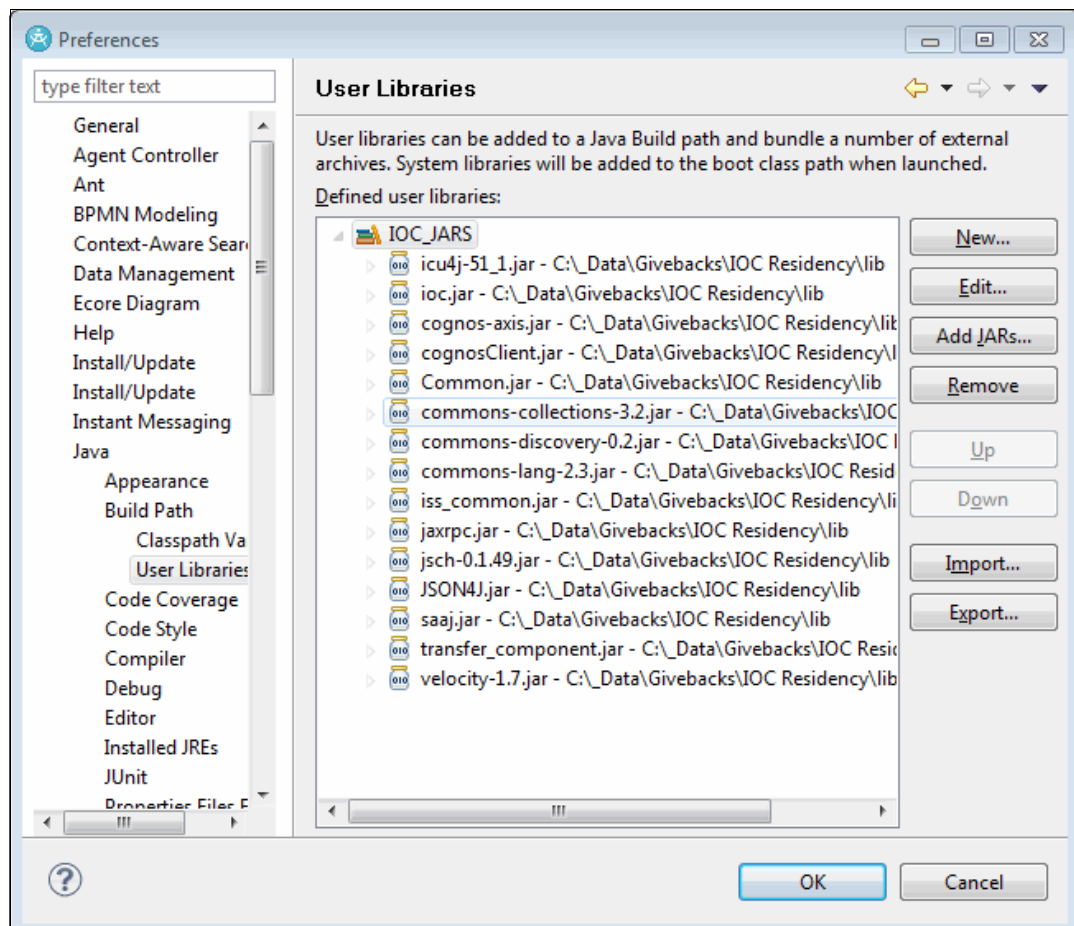


Figure 6-42 `IOC_JARS` user library

6. Import the sample portlet provided with this book. Click **File** → **Import** → **General** → **Existing Projects into Workspace** and click **Next**.
7. Click **Select Archive File** and click **Browse**. Select the sample named `sample_ioc_ui_custom_portlet`, which is provided with the book. Your window looks similar to Figure 6-43. Click **Finish**.

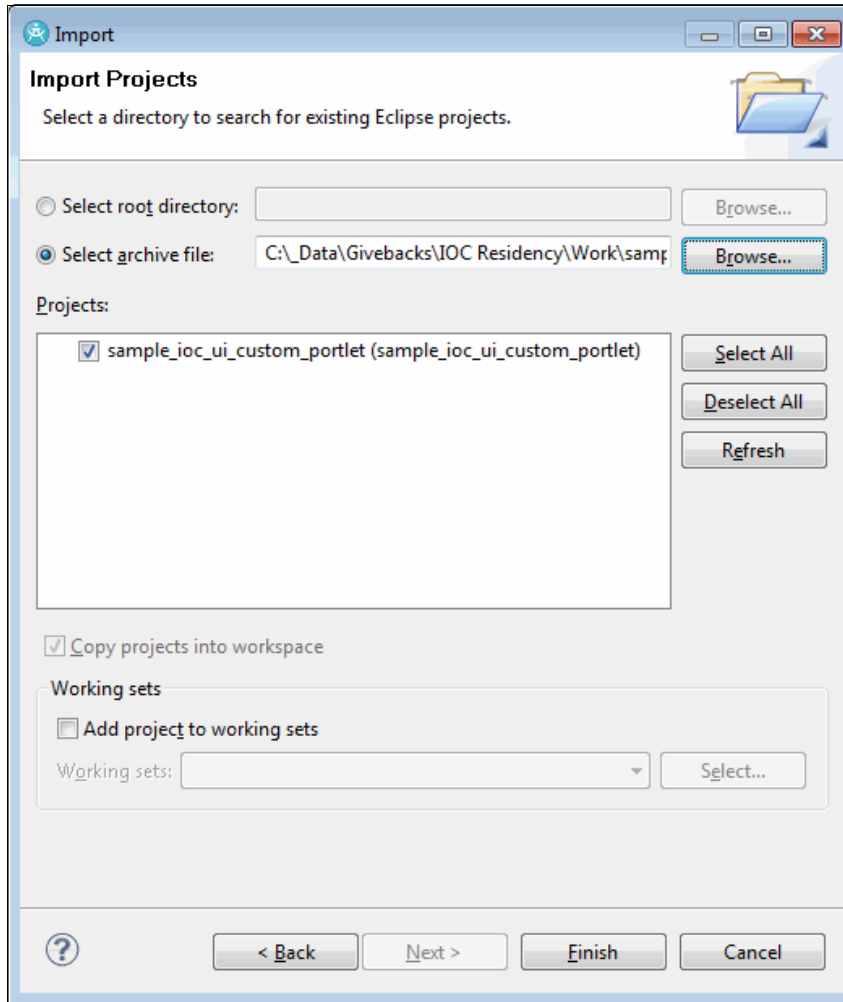


Figure 6-43 Importing the sample project

8. The project is now imported into the workspace. The project uses a user library to find the IBM Intelligent Operations Center JARS. If the user library created in step 3 on page 240 was named IOC_JARS as in this example, the project finds it automatically.

If the user library was named with another name, right-click the project and click **Properties**. Select **Java Build Path** and then select the **Libraries** tab. Edit the **IOC_JARS** library to use the correct name and click **OK**.

9. In the Enterprise Explorer view, expand **Java Resources** and open the **CustomPortlet.java** file as shown in Figure 6-44 on page 242.

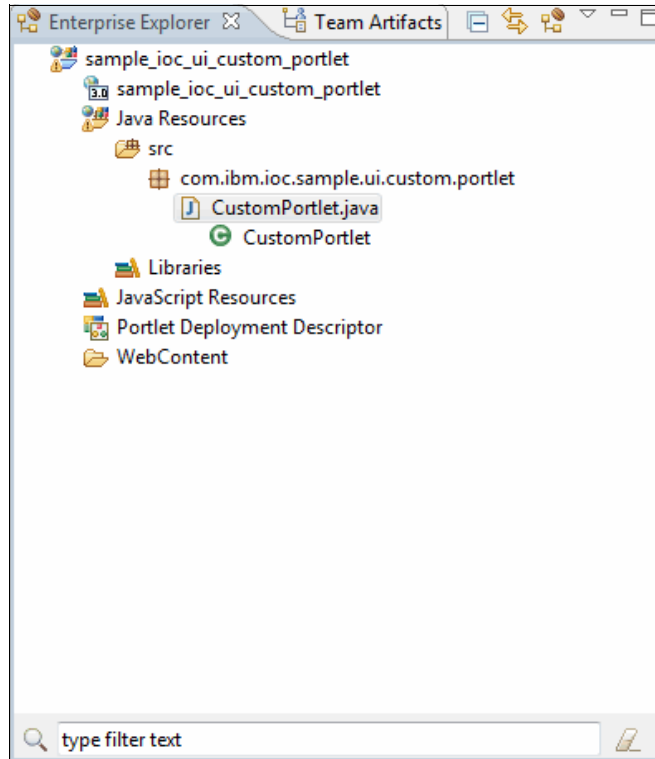


Figure 6-44 Opening CustomPortlet.java file from the Enterprise Explorer view

10. Inspect the CustomPortlet.java code. This code is a simple IBM Intelligent Operations Center custom portlet that prints a text string. The portlet extends the base IBM Intelligent Operations Center portlet called com.ibm.ioc.portlet.ISSAbstractPortlet.

6.6.2 Deploying custom portlets

Deploy a custom portlet to IBM Intelligent Operations Center:

1. Right-click the portlet project (**sample_ioc_ui_custom_portlet**) and click **Export** → **WAR file**. In the next dialog, type a name for the WAR file, and click **Finish**.
2. From the IBM Intelligent Operations Center solution interface, click **Administration** → **Portal Administration**, as shown in Figure 6-45.

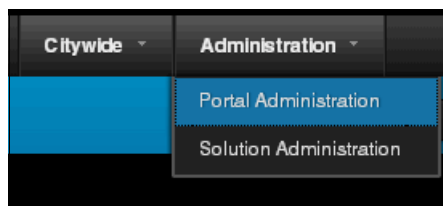


Figure 6-45 IBM Intelligent Operations Center - Portal Administration

3. Under Portlet Management, click **Web Modules** → **Install**. Click **Browse** and browse to the exported WAR file as shown in Figure 6-46 on page 243. Click **Next**.

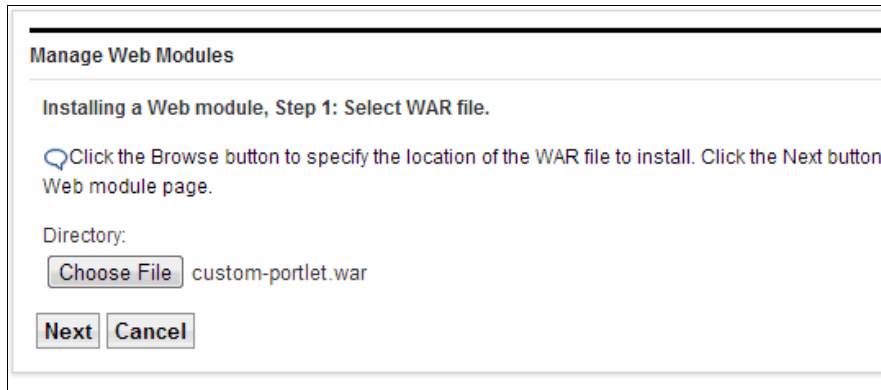


Figure 6-46 Installing a portlet

- The next window, Step 2: View WAR file contents, displays the WAR file content and gives you the option to select the Enterprise Application display name that will be deployed to WebSphere Application Server (see Figure 6-47). Take note of the Enterprise Application display name because it will be needed later and click **Finish**.

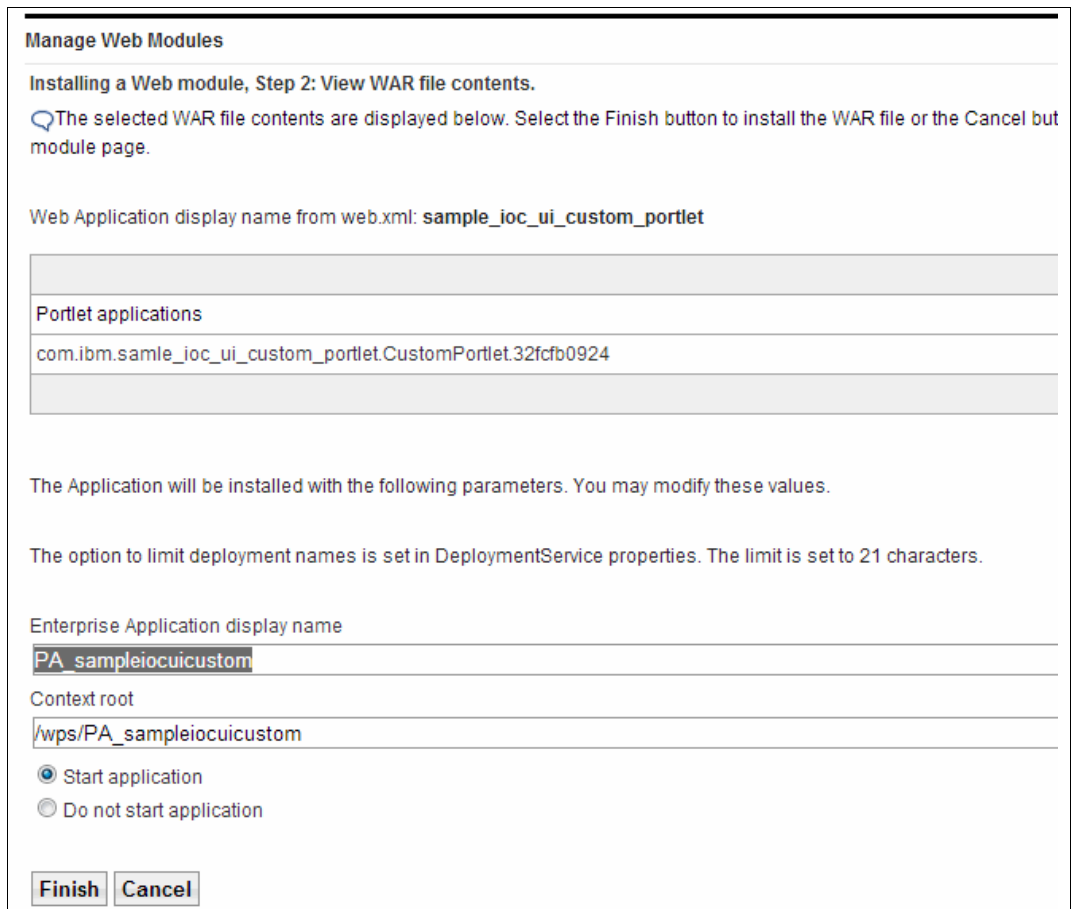


Figure 6-47 Installing a web module - Step 2: View WAR file content

The portlet is now deployed.

5. Specify the shared libraries that the portlet references.

You must configure the portlet to access the shared IBM Intelligent Operations Center JAR files:

- a. Log in to the WebSphere Application Server console on the application server with the administrative account. You can access the console from the IBM Intelligent Operations Center portal. Click **Administration** → **Solution Administration** → **Administration Consoles** → **Application server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Sametime Proxy).
- b. Click **Applications** → **Application Types** → **WebSphere enterprise applications**. Search for the portlet that was deployed in step 4 on page 243 with the Enterprise Application display name that you specified (PA_sampleiocuicustom in this example). Figure 6-48 shows the search results.
- c. Click the enterprise application name link (**PA_sampleiocuicustom**).

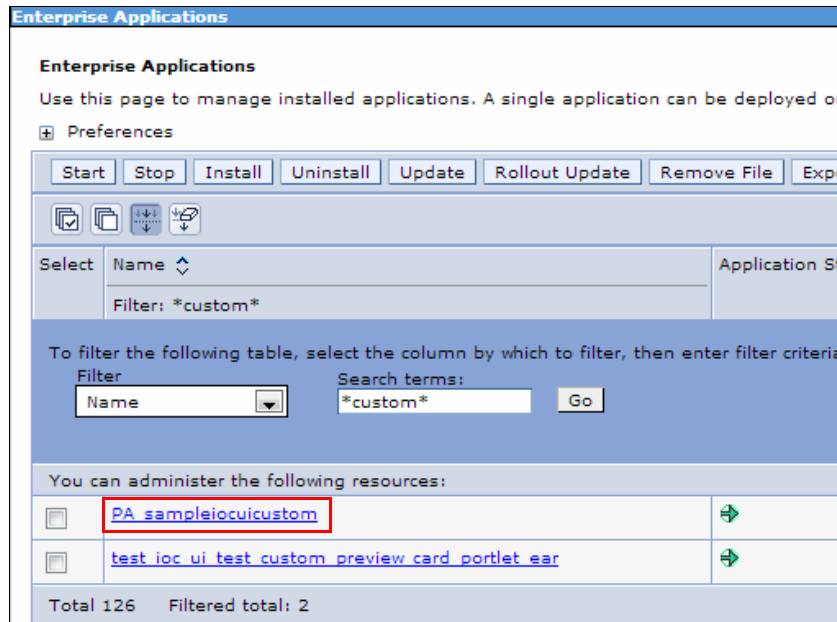


Figure 6-48 Portlet enterprise application in WebSphere Application Server

- d. Click **Shared library references**.
- e. In Shared library references, select the top-level enterprise application element as shown in Figure 6-49 and click **Reference shared libraries**.

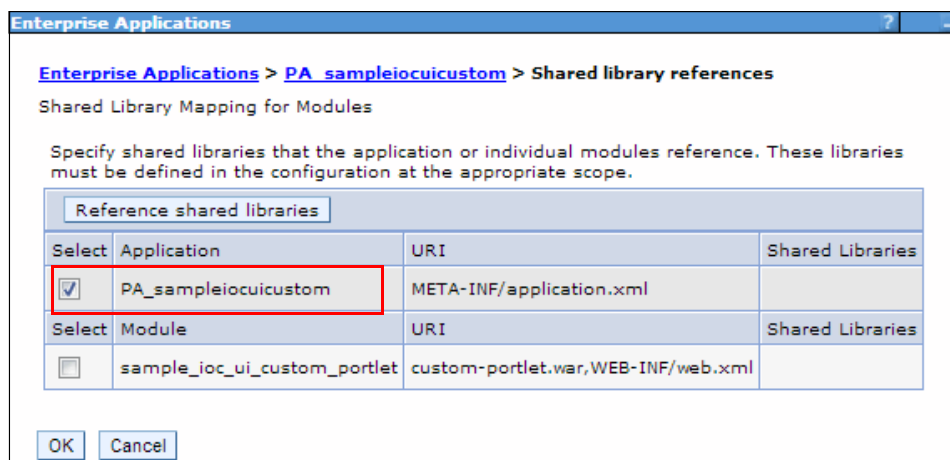


Figure 6-49 Configuring shared library

- f. In the Shared Library Mapping window, select **IOCCCommonJars** from the Available list and click the right arrow to move it to the Selected list as shown in Figure 6-50.

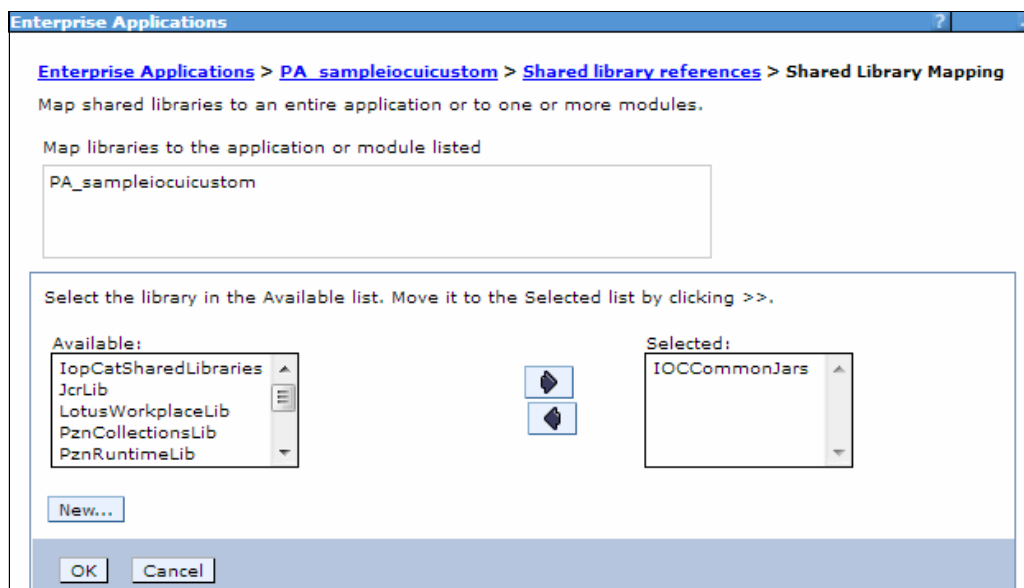


Figure 6-50 Selecting the IOCCCommonJars shared library

- g. Click **OK** twice.
 - h. A message appears reporting unsaved configurations. Click **Save**.
- At this point, the custom portlet is configured and ready to be added to a portal page.
6. Add the portlet to the portal page. Follow the steps described in “Adding custom portlets to the new page” on page 262 to add the portlet to the page.
 7. Open the portal page to display the custom portlet as shown in Figure 6-51 on page 246.

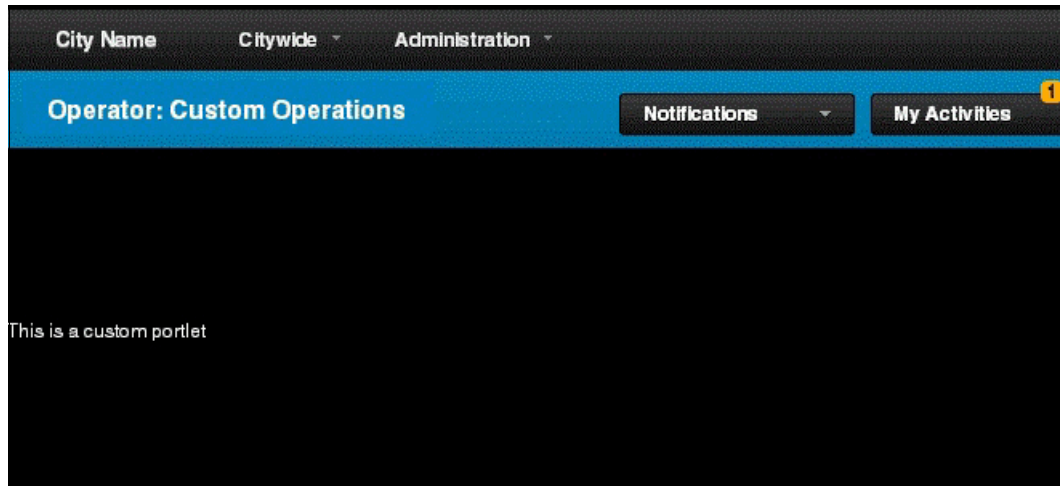


Figure 6-51 Custom portlet displayed on the IBM Intelligent Operations Center portal page

6.6.3 Using RestStore

RestStore is the common programming interface used by the IBM Intelligent Operations Center user interface components to call REST services. It also allows callers to detect data changes and update the UI. For more information about the purpose, benefits, and implementation of this feature, see 1.4, “RestStore” on page 13.

This section focuses on a usage example and illustrates how to use RestStore in a custom portlet in the IBM Intelligent Operations Center user interface. The complete code for this example is available with the sample provided with this book in the file `sample_ioc_ui_reststore_portlet`.

RestStore usage example

This example shows how to create a custom portlet that uses RestStore to display the existing filter panes and listen to events for changes in the filter panes, and refresh the view.

The sample RestStore uses the REST service `/ibm/ioc/api/filter-service/panels`.

Follow these steps:

1. Create the RestStore with the target service `/ibm/ioc/api/filter-service/panels` as shown in Example 6-16. This RestStore will refresh every 150 seconds and will not automatically start polling.

Example 6-16 Creating filter panel RestStore

```
panelStore = new RestStore({ // create the RestStore
    target: "/ibm/ioc/api/filter-service/panels",
    refreshInterval: 15000,
    autoStartPolling: false
});
```

2. Create a listener for add, delete, and update events to be able to refresh the view when the events happen. Example 6-17 on page 247 shows how to use the RestStore event framework to add listeners.

Example 6-17 Adding event listeners

```
panelStore.on("add", function(items) { // connect to events
    refreshUI();
});
panelStore.on("delete", function(ids) {
    refreshUI();
});
panelStore.on("update", function(items) {
    refreshUI();
});
```

3. Call the `start()` method. When the store is ready for the first time, populate the view as shown in Example 6-18.

Example 6-18 Polling the data from RestStore and refreshing the view

```
panelStore.start();// start polling
when(panelStore.ready, // wait for store to finish first poll interval
    function() {
        refreshUI();
    }
);
```

4. Import the sample code provided in the file `sample_ioc_ui_reststore_portlet` into the IBM Rational Application Developer workspace.
5. Follow the steps described in 6.6.2, “Deploying custom portlets” on page 242 to create a new page and deploy the sample to the IBM Intelligent Operations Center user interface. For example, create a page with the name `Filter Panes` by clicking **Content Root** → **City Name** → **Citywide** and add the `sample_ioc_ui_reststore_portlet` to that page.

Test the RestStore portlet:

1. Navigate to the newly created page, for example, select **Citywide** → **Filter Panes**. The initial view of the sample RestStore portlet is shown in Figure 6-52. Note the colors assigned to the different filter panes; the Events pane is set to green.

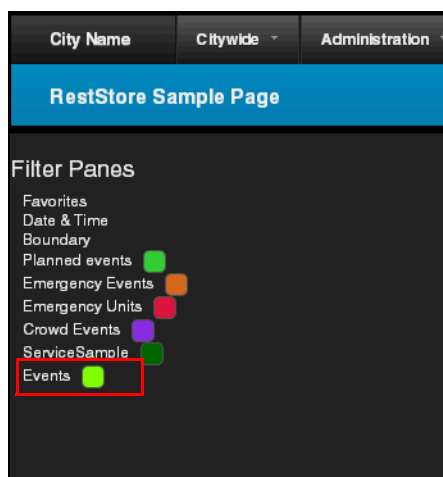


Figure 6-52 RestStore portlet showing Filter Pane colors

2. Assign a different color to the Events filter pane.
Click **System Administration** → **Configuration Tools** → **Filter Panel**. The RestStore Portlet updates the view on the next polling interval (within 15 seconds).
3. Go back to the page with the custom RestStore portlet and notice that the colors are updated automatically to match the changes on the filter pane as shown in Figure 6-53.

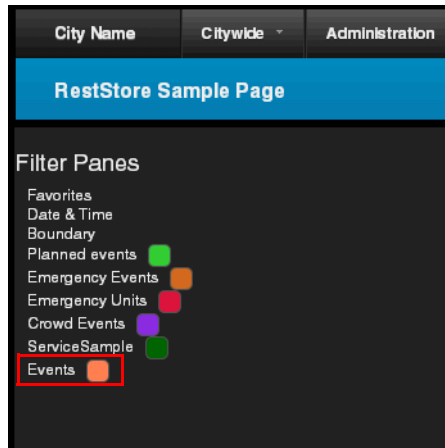


Figure 6-53 RestStore portlet updated to show new color for Events pane

6.6.4 Using the translation widget and internationalization service

The purpose of the internationalization service is to translate text strings, such as labels, used in the IBM Intelligent Operations Center UI components based on the user's locale. This feature is useful in geographic regions that have multiple official languages and require multi-language support in all software text.

The REST API service name is an *i18n-service* (i18N is an abbreviation for *internationalization* where 18 stands for the number of letters between i and n).

Developers can use the *translation widget* in custom JavaScript elements in the IBM Intelligent Operations Center UI components, such as portlets and widgets.

The translation widget extends the `idx.form.TextBox` widget, and all properties and functions defined in `IDX TextBox` are available in the translation widget. The additional functions are related to loading and saving the following values:

- ▶ `group`: The resource collection name
- ▶ `locale`: The locale of the value
- ▶ `key`: The locale independent identifier (do not use "." in the key value)
- ▶ `value`: The translated value

The translation widget also uses the `RestStore` class (see 6.6.3, "Using RestStore" on page 246) to call the `/ibm/ioc/api/i18n-service/resources` REST service to manage the `group`, `locale`, `key`, and `value` values.

More details about the `IDX TextBox` widget are available in the *Dojo documentation* at this website:

<http://dojotoolkit.org/reference-guide/1.9/dijit/form/TextBox.html>

In the UI, the internationalization widget provides a translation icon within the text box and a tooltip dialog to enter the values for the supported locales. The tooltip dialog opens when the user clicks it or presses the spacebar key.

The translation widget always shows the text in the locale configured for the user. The `SupportedLocales` system property defines the order and the set of supported locales.

To modify the system property value, log in to the IBM Intelligent Operations Center portal and click **Administration** → **Solution Administration** → **System Administration** → **System Properties** → **SupportedLocales**.

The user's locale is defined by the configuration in the user profile or the web browser language settings, or it is taken from the default configuration if it is not configured in the user profile or browser.

Note: If there are no supported locales, the translation widget does not show either the translation icon or the tooltip window. It appears as a standard text box and the value entered by the user is saved as the *default* locale for the group and key.

Using the translation widget

The translation widget is defined in the `ioc/com/ibm/ioc/widget/I18nTextBox` package and automatically available to any Dojo code. The Dojo code must run in a portlet in one of the IBM Intelligent Operations Center solution themes.

Example 6-19 shows how to construct the translation widget. Table 6-5 on page 250 lists the translation widget properties.

Example 6-19 Constructing a translation widget to store and retrieve a custom value

```
i18nWidget = new I18nTextBox({
    id: "unique_id_i18n",
    group: "SampleI18n",
    key: "label",
    fieldWidth: "300px",
    maxLength: 250,
    required: true,
    label: "Label:",
    labelAlignment: "vertical",
    help: "Help goes here"
}, "<portlet:namespace/>_i18n");
```

Translation widget properties

The translation widget has a number of properties in addition to those that are available in the `i18n-service`. The additional properties are specified in Table 6-5 on page 250.

Table 6-5 Translation widget properties

Property	Description	Example
visibleText	Locale-specific text that is visible in the translation widget text box on the user interface. See usage example in "Translation widget usage example" on page 250.	"Some sample text"
fieldWidth	Specifies the width of the field in pixels (px) or other unit supported by CSS.	fieldWidth:"300px"
maxLength	Maximum length of the stored string in characters.	maxLength: 250
required	Boolean value. It defines whether the field is required.	required: true
readOnly	Boolean value. It defines whether the field is read only.	readOnly: true
label	String value. It contains the field label (the i18n-service can be used for localization).	label: "Label:"
labelAlignment	String value. It specifies the label alignment.	labelAlignment: "vertical"
help	String value. It contains the help text displayed in the icon with a question mark immediately after the field label (the i18n-service can be used for localization).	help: "Help goes here"

Translation widget functions

Table 6-6 summarizes the translation widget functions.

Table 6-6 Translation widget functions

Function	Description
load	Loads the values for a group and key for all locales.
save	Saves the values in the translation widget.

Translation widget usage example

The portlet in this example includes the translation widget. The translation widget saves the value entered by the user. The same translation widget can be used to retrieve the saved value and display it in a JavaScript alert in a pop-up window.

There are two buttons that interact with the widget:

- ▶ The first button saves or updates the value entered by the user.
- ▶ The second button displays the value in a pop-up window.

Example 6-19 on page 249 shows how to construct this sample translation widget.

Example 6-20 shows how to define the button that saves the value entered by the user into the locale store.

Example 6-20 Defining the button to save the value entered by the user

```
// The Button class used in this example is implemented in "dijit/form/Button"

saveButton = new Button({
    label: "Save",
    tabindex: 0,
    onClick: function() {
// The i18nWidget is defined in Example 6-19 on page 249
        i18nWidget.save();
    }
}, "<portlet:namespace/>_saveButton");
```

Use the get function to display the saved value as shown in Example 6-21.

Example 6-21 Defining the button that displays the saved value in a pop-up window

```
displayButton = new Button({
    label: "Display",
    tabindex: 0,
    onClick: function() {
        alert("Visible text: " + i18nWidget.get("visibleText"));
    }
}, "<portlet:namespace/>_displayButton");
```

The complete source code for this example is available in the samples file `sample_ioc_ui_i18n_portlet`.

Import the sample code into the IBM Rational Application Developer workspace and follow the steps described in 6.6.2, “Deploying custom portlets” on page 242 to deploy the sample translation widget to IBM Intelligent Operations Center. Figure 6-54 shows the results displaying the sample portlet that includes the translation widget.

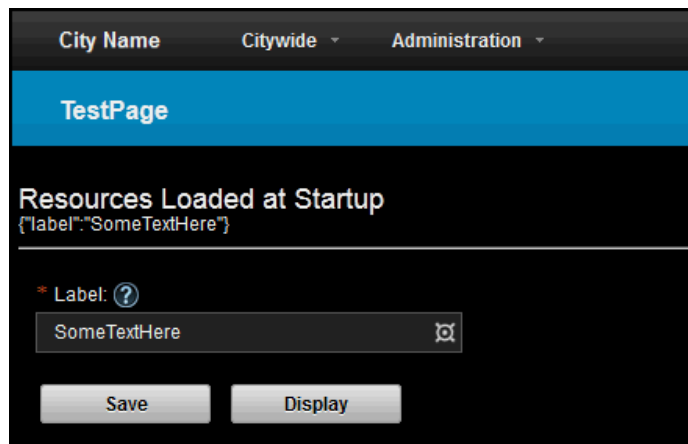


Figure 6-54 Custom portlet with translation widget

6.7 Adding custom help links

The following steps are used to add custom help links:

1. Add an Eclipse plug-in with the custom help articles.
2. Create a new Help menu item and add it to the Help menu.

The sample used in this section is `com.ibm.ioc.custom.doc_1.6.0.zip`.

6.7.1 Adding an Eclipse plug-in with custom help articles

Follow these steps:

1. Navigate to the WebSphere Application Server administrative console and log in with the administrative user account.

From the IBM Intelligent Operations Center portal interface, navigate to **Solution Administration** → **System administration** → **Administrative consoles** → **Application server** (WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal, IBM Business Monitor, IBM Worklight, IBM Cognos Business Intelligence, IBM ILOG CPLEX, and Sametime Proxy).

2. Stop the portal cluster:
 - a. Click **Servers** → **Clusters** → **WebSphere application server clusters**.
 - b. Select **PortalCluster**.
 - c. Click **Stop**.
3. Upload the sample Eclipse plug-in provided with this book (file name `com.ibm.ioc.custom.doc_1.6.0.zip`) to the application server.
4. Unpack the plug-in in this directory:

```
/opt/IBM/WebSphere/wp_profile/installedApps/cell1/iss_help_war.ear/ibm_cities_help.war/WEB-INF/eclipse/plugins
```
5. Change the plug-in permissions to be owned by `ibmadmin` and group `ibmadmins`. For this example, run the `chown -R ibmadmin:ibmadmins com.ibm.ioc.custom.doc_1.6.0` command from `opt/IBM/WebSphere/wp_profile/installedApps/cell1/iss_help_war.ear/ibm_cities_help.war/WEB-INF/eclipse/plugins` directory.
6. Restart the portal cluster from the WebSphere Application Server administrative console:
 - a. Click **Servers** → **Clusters** → **WebSphere application server clusters**.
 - b. Select **PortalCluster**.
 - c. Click **Start**.

6.7.2 Creating a new Help menu item and adding it to the Help menu

The new Help menu item is added from a custom widget that can be a part of a custom portlet or an extension for a standard portlet.

To create a new Help menu item, the `registerPortlet(args)` function registers the widget running inside the portlet within the theme and publishes the values for the following properties to the Dojo topic named `/ibm/ioc/theme/help`:

- ▶ `label`: Label for the custom help menu item
- ▶ `help`: Link to the HTML file for the help topic

- ▶ `type`: Type of help plug-in. By default, it is set to `ioc` and links to the IBM Intelligent Operations Center help plug-in (`com.ibm.ioc.doc`). These other values are valid:
 - `icpo`: Links to IBM City Planning and Operations help plug-in (`com.ibm.icpo.doc`)
 - `trans`: Links to IBM Intelligent Transportation help plug-in (`com.ibm.transport.doc`)
 - `water`: Links to IBM Intelligent Water help plug-in (`com.ibm.water.doc`)
- ▶ `customHelp`: Boolean property that, if specified, uses the `help` parameter as the entire help doc path.

Note: Use `Library.js` from the IBM Intelligent Operations Center solution theme that contains the set of common JavaScript functions of the system. In a default installation, it is in this directory:

```
/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_theme_ear.ear/ioc_theme_s
tatic_js_web.war/js/com/ibm/ioc/
```

Example 6-22 shows the sample code that adds the custom Help menu item *Custom app* to the Help menu.

The sample code links to the file `com.ibm.ioc.custom.doc/html/custom_app.html` with the custom help article that is part of the Eclipse plug-in installed as described in 6.7.1, “Adding an Eclipse plug-in with custom help articles” on page 252.

Note: For the full listing of the sample code, see the custom widget for the content panel described in 6.2, “Customizing the content panel” on page 197.

Example 6-22 Adding a custom help menu item from a custom widget

```
define(
  [
    // [...]
    "ioc/com/ibm/ioc/Library"
  ],
  function(/*[...]*/, Library) {
// [...]
/**
 * Function to configure the help in the help drop down.
 */
    _configureHelp: function(){
// Logging function on entry goes here
      this._library.registerPortlet({
        label: "Custom app",
        help: "com.ibm.ioc.custom.doc/html/custom_app.html",
        customHelp: true
      });
// Logging function on exit goes here
    },
// [...]
});
```

The Help menu includes the customized Help menu item as shown in Figure 6-55 on page 254.

For detailed instructions to customize the content viewer portlet and add it to a new page, see 6.2, “Customizing the content panel” on page 197.

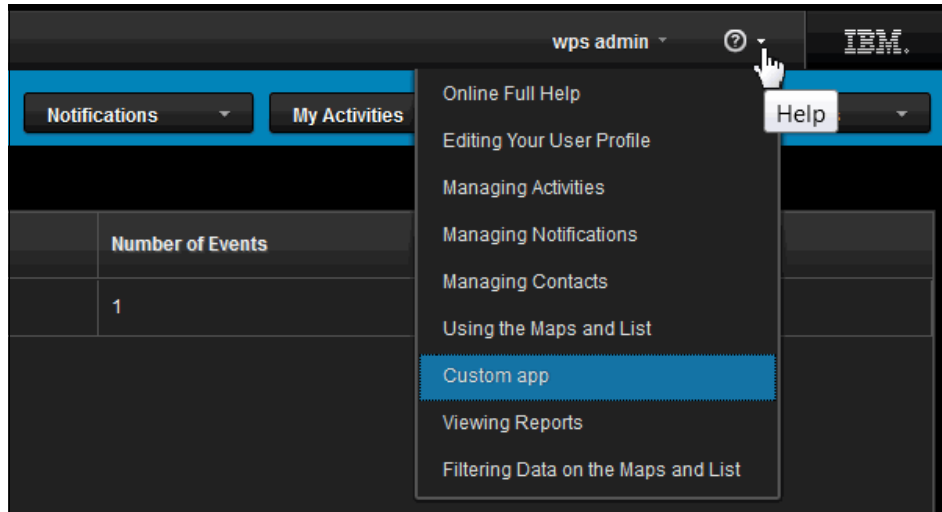


Figure 6-55 New custom help menu item added from the widget

Developers can also publish the values directly to the Dojo topic as shown in Example 6-23.

Example 6-23 Publishing details for custom help menu item to the Dojo topic

```
define(  
  [  
    // [...]  
    "dojo/topic"  
  ],  
  function(/*[...]*/, topic) {  
// [...]  
  topic.publish("/ibm/ioc/theme/help", {  
    label: "Custom app",  
    help: "com.ibm.ioc.custom.doc/html/custom_app.html",  
    customHelp: true  
  });  
  
  },  
// [...]  
});
```

Note: When creating a customHelp help link, it is not required to specify both type and customHelp. The type is used to determine the help documentation path. If customHelp is specified as true, type is ignored and the help property specifies the documentation path. If type is specified as ioc, it is assumed that the documentation path is not provided and that the default path com.ibm.ioc.doc needs to be used.

6.8 Customizing the More Actions menu

You can define custom actions that are displayed as available actions on the Actions tab in the data source configuration tool. If you assign a custom action to a data source, it is displayed in the More Actions menu that is on the preview card for the data source.

The More Actions menu shows the list of options that are available for the user to interact with the data currently displayed on the window.

For example, for the Map tab of the content panel, the options in the More Actions menu allow you to create a new item, change the base map layer, or reset the map.

These options can be customized. They are context-specific, which means that the More Actions menu can display a different set of options depending on the currently active widget in the portlet displayed on the page.

In Example 6-24, the custom Dojo widget extends the `com.ibm.ioc.contentviewer.View` library and implements the `getMoreActionsMenuItems()` function to add the custom menu option `Test More Action Feature` associated with the callback function `_moreActionsFunction()`.

The callback function can be used to implement the custom action and call the back-end system or display another widget with interaction options for the user.

Note: The custom widget in 6.2, “Customizing the content panel” on page 197 contains the full example.

Example 6-24 Adding a custom option to the More Actions menu from a Dojo widget

```
define([
    // [...]
    "ioc/com/ibm/ioc/contentviewer/View"
],
function(/*[...]*/ View) {

    return declare("test.contentviewerWidget", [View, /*[...]*/], {
/**
 * Return the custom option for the More Actions menu
 */
        getMoreActionsMenuItems: function(){
            // Logging on function entry goes here

            var menuItems = {};

            menuItems.newIoCListItem = {
                label: 'Test More Action Feature',
                callback: lang.hitch(this, this._moreActionsFunction)
            };

            // Logging on function exit goes here
            return menuItems;
        },
/**
```

```

* Implement the callback function - just displaying an alert
*/
    _moreActionsFunction: function(){
        alert('More actions callback function being called');
    }
// [...]
}
);
});

```

Custom menu options can also be edited by publishing the array of items to the following Dojo topics:

- ▶ /ibm/ioc/more_actions/add: To add custom menu options
- ▶ /ibm/ioc/more_actions/remove: To remove custom menu options

Example 6-25 shows the details.

Example 6-25 Editing custom options in the More Actions menu by publishing to Dojo topics

```

define(
    [
        // [...]
        "dojo/topic"
    ],
    function(/*[...]*/, topic) {
        // [...]
/**
 * Adding option called "Test More Action Feature" and callback function
 * _moreActionsFunction by publishing to the topic "/ibm/ioc/more_actions/add".
 */
        var moreActions = [];
        moreActions["demo_action"] = {
            label: "Test More Action Feature",
            callback: this._moreActionsFunction
        };
        topic.publish("/ibm/ioc/more_actions/add", moreActions);
        // [...]
/**
 * Removing option called "Test More Action Feature" by publishing to the topic
 * "/ibm/ioc/more_actions/remove".
 */
        var removeList = [];
        removeList.push("demo_action");
        topic.publish("/ibm/ioc/more_actions/remove", removeList);
        // [...]
/**
 * Implement the callback function - just displaying an alert
 */
        _moreActionsFunction: function(){
            alert('More actions callback function being called');
        }
        // [...]
    });

```

Figure 6-7 on page 197 shows the results.

6.9 Creating custom solution pages

This section describes how to create a new solution page and add a custom portlet to it. Creating new portal pages and deploying custom portlets enable you to create a new custom user interface for a new user role or add a new custom portlet with a new set of features for existing roles.

For example, use this approach to add a new Cognos portlet with custom reports or a new custom portlet to integrate with third-party systems.

In this section, the copy of the content viewer portlet is used as an example of a custom portlet. For information about creating a copy of the content viewer portlet, see “Copying the standard content viewer portlet and modifying the configuration” on page 212.

6.9.1 IBM Intelligent Operations Center solution themes

IBM Intelligent Operations Center comes with three custom themes for WebSphere Portal:

- ▶ IOC Portal 8 Admin Theme: Used in the Solution Administration pages
- ▶ IOC Portal 8 Theme: Used in all other pages, such as Operations or Status
- ▶ IOC Portal 8 Reports Theme: Used for displaying Cognos portlets with custom reports

The IOC Portal 8 Admin Theme is used only on the Solution Administration pages and cannot be used for the other pages in the system. Its structure is similar to the IOC Portal 8 Theme, except for the left content panel (number 1 on Figure 6-56), which is used only for navigation between Solution Administration pages.

The difference between IOC Portal 8 Theme and IOC Portal 8 Reports Theme is only in the number and location of panels.

The default IOC Portal 8 Theme has four panels. Each of the panels maps to a certain part of the user interface; therefore, the portlets that are added to this panel appear in certain areas of the page.

Figure 6-56 and Figure 6-57 on page 258 show the mapping between the IBM Intelligent Operations Center solution page elements and the panels in the Portal Administration view.

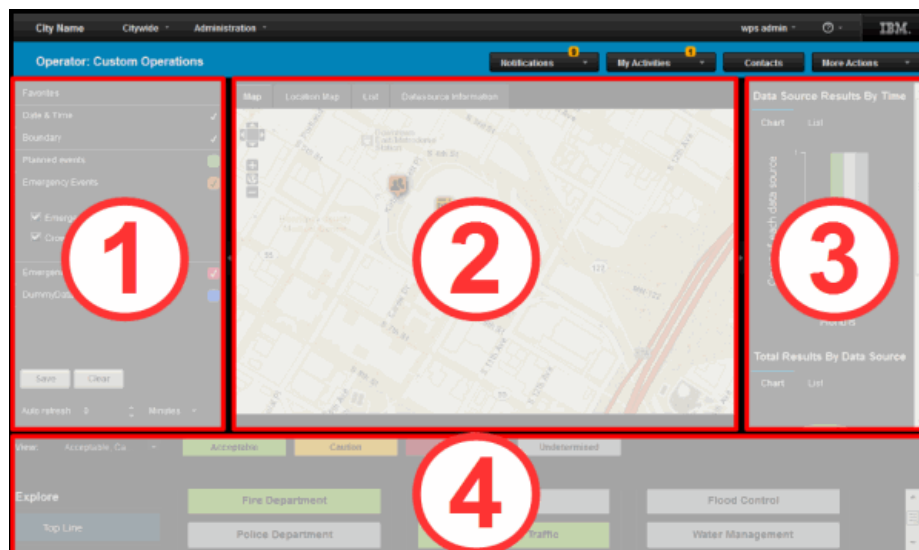


Figure 6-56 Page elements in the IOC Portal 8 Theme

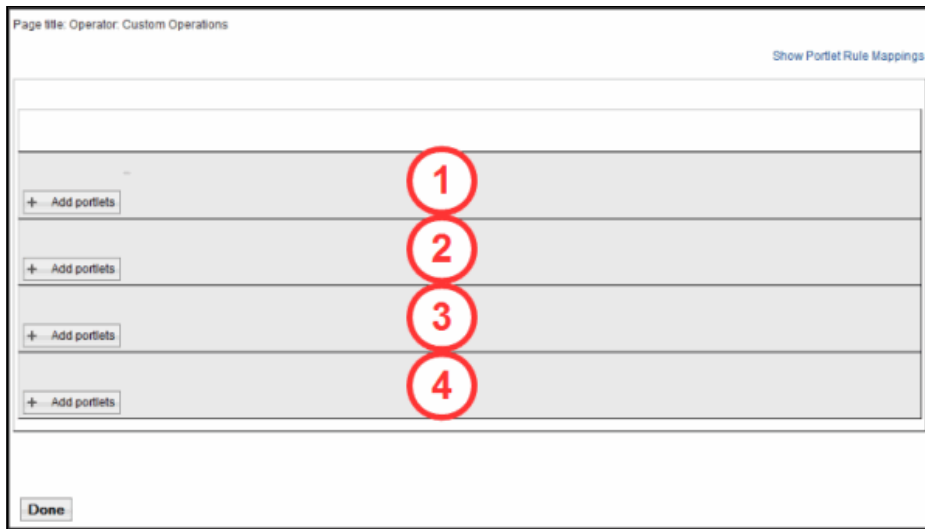


Figure 6-57 Panels for the IOC Portal 8 Theme in the Portal Administration interface

The IOC Portal 8 Reports Theme has two panels and can be used for Cognos portlets with custom reports and any other portlets if this design is more suitable for some reason. See mapping between the panels and the views on Figure 6-58 and Figure 6-59 on page 259.

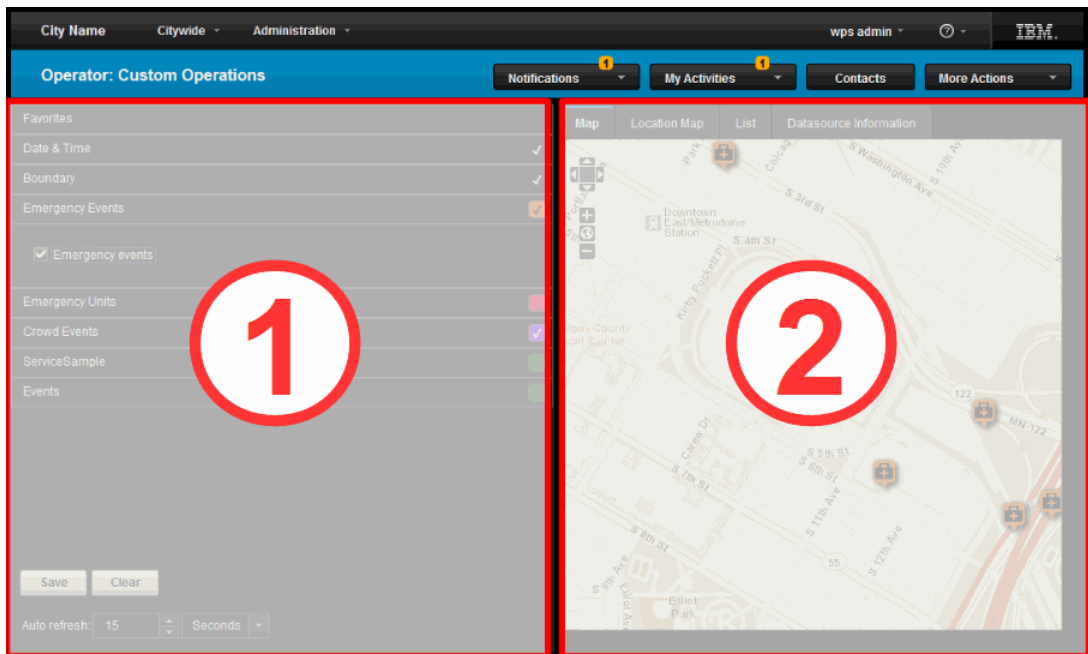


Figure 6-58 Page elements in the IOC Portal 8 Reports Theme

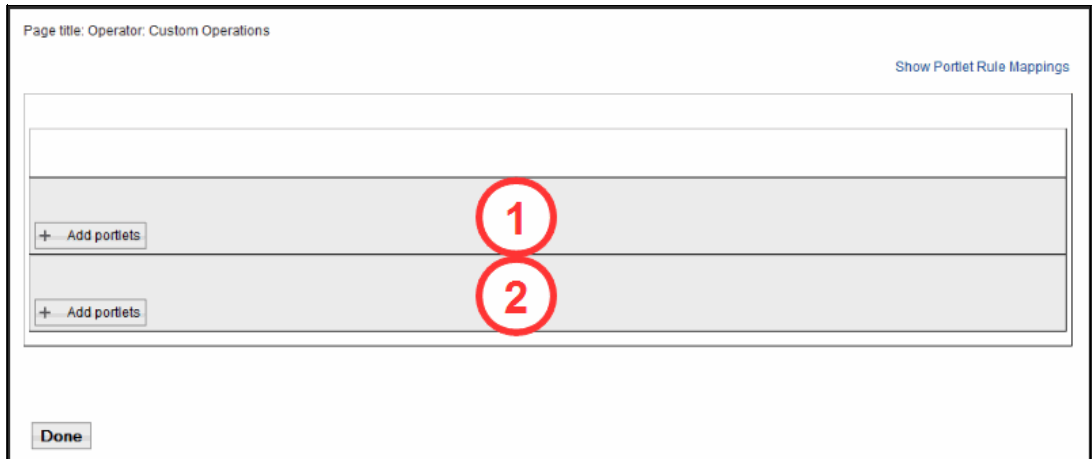


Figure 6-59 Portlet panels for IOC Portal 8 Reports Theme in the Portal Administration interface

6.9.2 Creating custom portal pages

The following steps summarize the steps to create custom portal pages and add a custom portlet to the page:

1. Navigate to the Portal Administration view and create a new page.
2. Add custom portlets to the new page.
3. Test the new portal page and portlets.

Navigate to the Portal Administration interface and create a new page

Follow these steps:

1. Click **Administration** → **Portal Administration** (Figure 6-60).

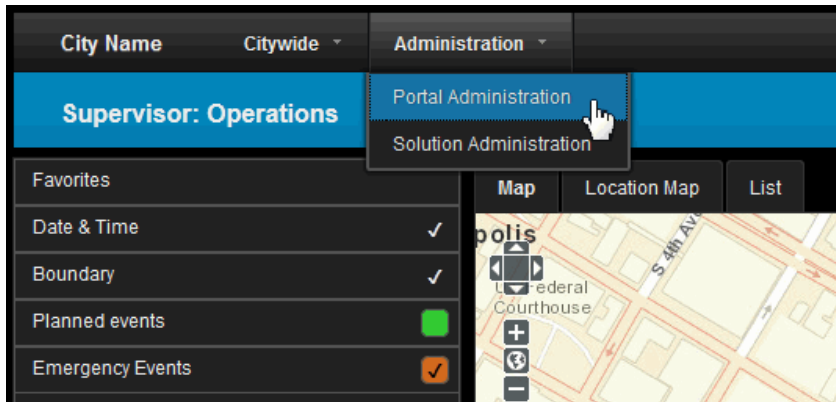


Figure 6-60 Navigating to Portal Administration interface

2. Click **Portal User Interface** → **Manage Pages** (see Figure 6-61).

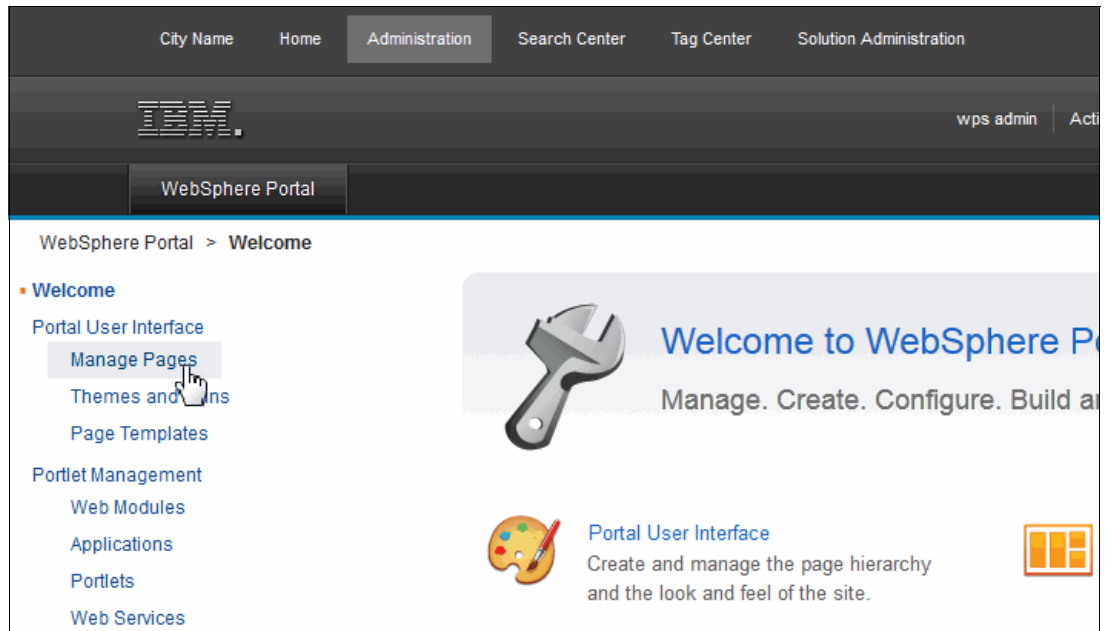


Figure 6-61 Accessing the Manage Pages page

3. Navigate to **Content Root** → **City Name** → **Citywide** and click **New Page** (see Figure 6-62).

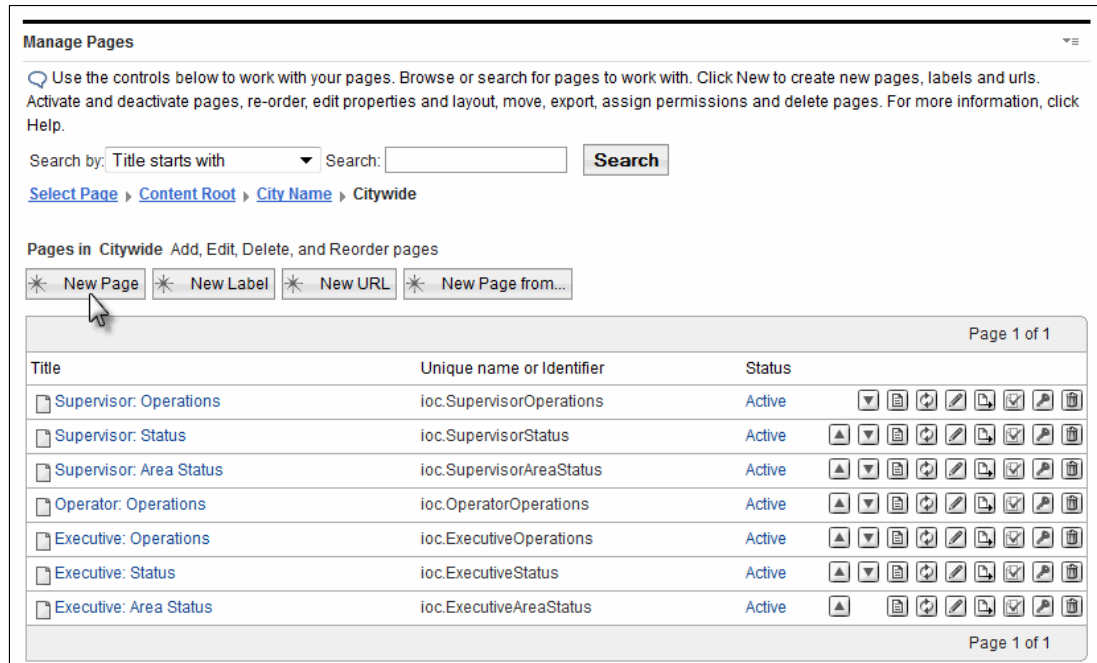


Figure 6-62 Creating a new portal page

4. Enter the name for the new page, for example, Operator: Custom Operations, and click **OK** (see Figure 6-63).

New page: Citywide

Use the controls below to work with your pages to specify your page properties. Expand Options to choose

Title:
Operator: Custom Operations

Unique Name:
[Empty text box]

Note: If the unique name you entered for this page already exists, it will not be created or updated.

Friendly URL name:
[Empty text box]

Theme:
-----Inherit Parent Theme-----

Icon:
[Empty text box]

I want to make this page my private page

Aggregation - Render Mode:
This setting will revert to SSA during runtime if the theme assigned with the page does not support CSA.

Inherit Parent Render Mode
 Client Side Aggregation - Rendering
 Server Side Aggregation - Rendering

Page Properties

Type of Page

Page Cache Options

Figure 6-63 Entering a new page title

Adding custom portlets to the new page

Follow these steps:

1. For the new page, click the **Edit Page Layout** icon (highlighted in Figure 6-64).

The screenshot shows the 'Manage Pages' interface. At the top, there is a search bar with a dropdown menu set to 'Title starts with' and a 'Search' button. Below the search bar is a breadcrumb trail: 'Select Page > Content Root > City Name > Citywide'. Underneath, there are buttons for 'New Page', 'New Label', 'New URL', and 'New Page from...'. The main part of the interface is a table with the following columns: 'Title', 'Unique name or Identifier', 'Status', and a set of action icons. The table contains several rows of page entries, all with a status of 'Active'. The last row, 'Operator: Custom Operations', is highlighted with a red box. The 'Edit Page Layout' icon (a pencil) in the action column for this row is also highlighted with a red box and has a mouse cursor pointing to it. The page number 'Page 1 of 1' is visible in the top right corner of the table area.

Title	Unique name or Identifier	Status	
Supervisor: Operations	ioc.SupervisorOperations	Active	[Icons]
Supervisor: Status	ioc.SupervisorStatus	Active	[Icons]
Supervisor: Area Status	ioc.SupervisorAreaStatus	Active	[Icons]
Operator: Operations	ioc.OperatorOperations	Active	[Icons]
Executive: Operations	ioc.ExecutiveOperations	Active	[Icons]
Executive: Status	ioc.ExecutiveStatus	Active	[Icons]
Executive: Area Status	ioc.ExecutiveAreaStatus	Active	[Icons]
Operator: Custom Operations	id:Z6_J0KC0101JG3010A8IM5QI80006	Active	[Icons]

Figure 6-64 Going to page customization interface

2. Click the first **Add portlets** option to add portlets to be displayed on the left side of the window (see Figure 6-65). For more information about page layout, see 6.9.1, “IBM Intelligent Operations Center solution themes” on page 257.

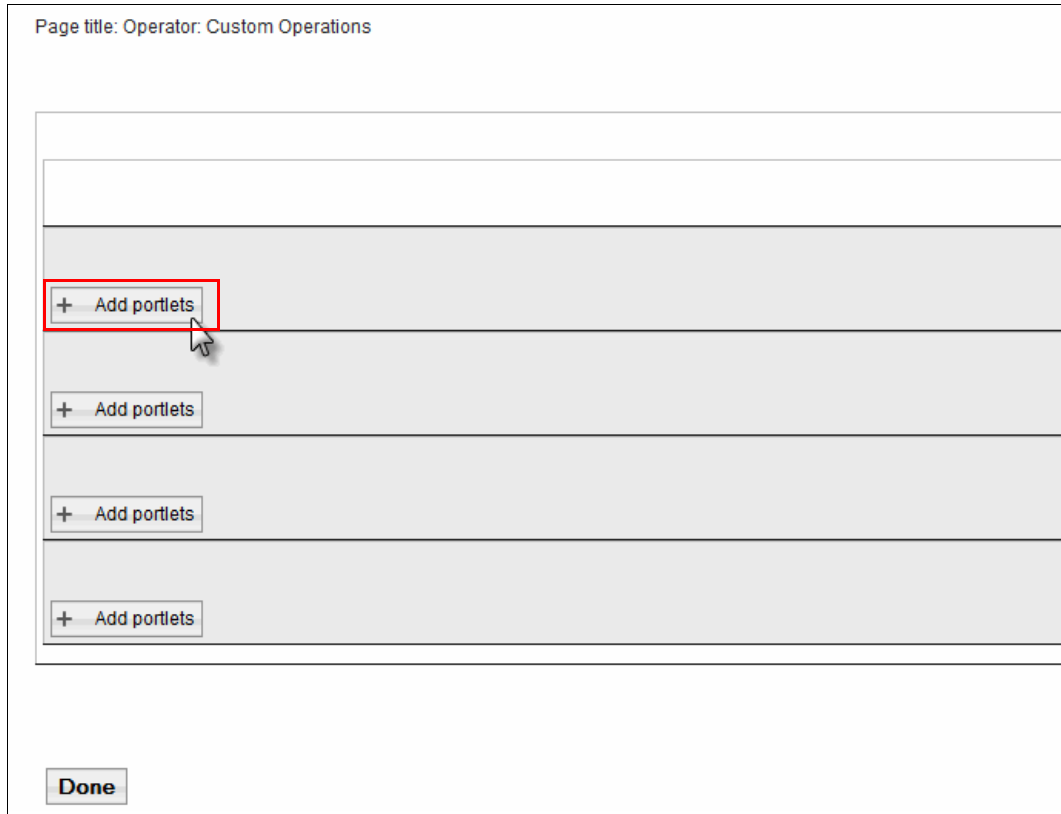


Figure 6-65 Adding a new portlet to the page

3. In the Search field, enter `ioc_ui_filter_portlet` and click **Search**.
4. Select the **ioc_ui_filter_portlet** portlet, and click **OK** (see Figure 6-66).

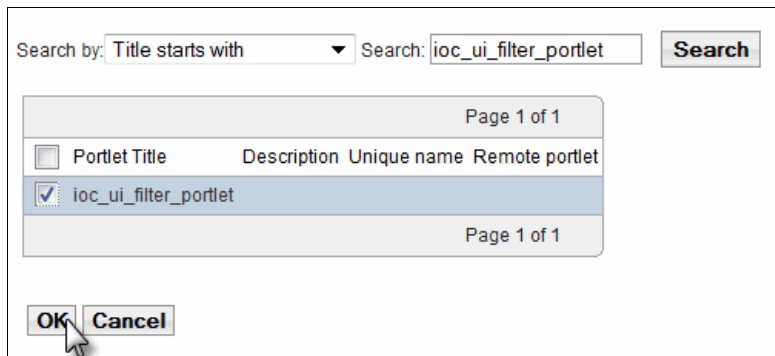


Figure 6-66 Selecting the portlet to add to the page

- Repeat steps 1 through 4 to add the `ioc_content_viewer_portlet_custom` (the portlet to be displayed in the center of the portal page) and `ioc_ui_charting_portlet` (the portlet to be displayed on the right side of the portal page). Figure 6-67 shows the results.

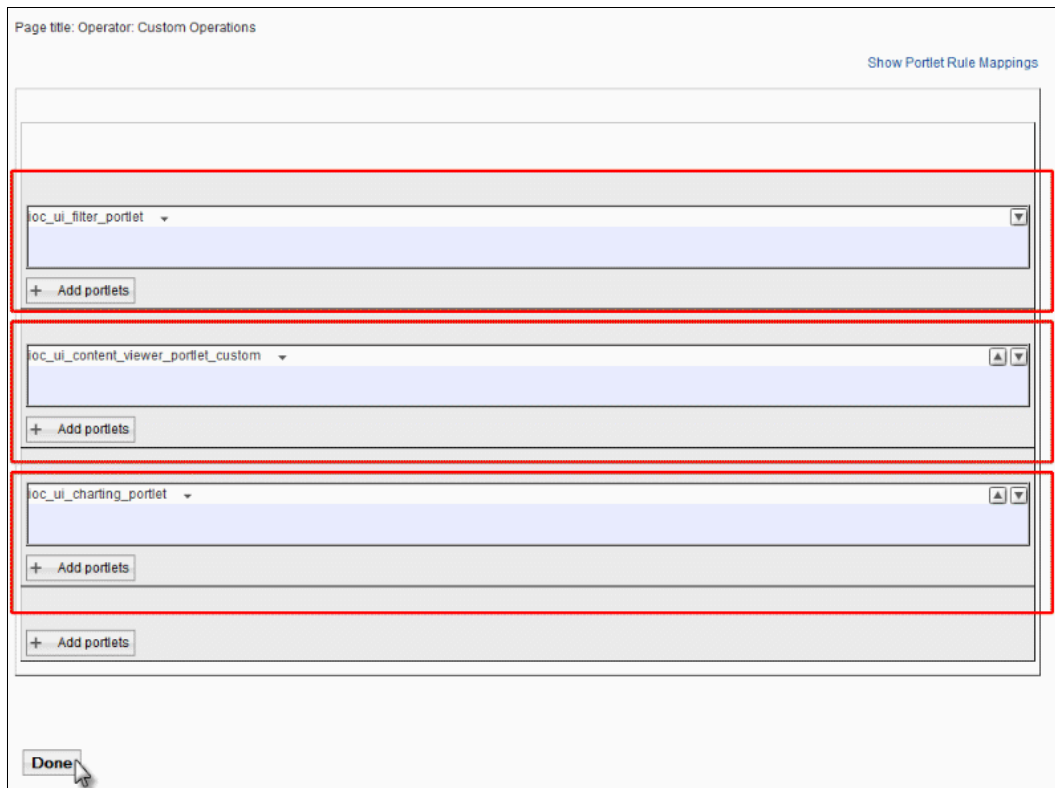


Figure 6-67 Page configuration with portlets added

Testing the new portal page and portlets

- Click the **City Name** link on the upper-left corner of the page (see Figure 6-68).

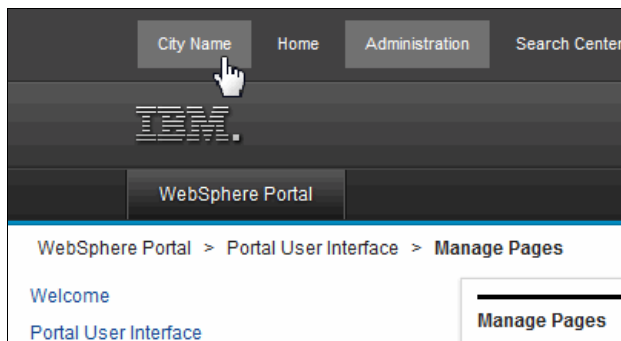


Figure 6-68 Navigating back to the user interface

2. Navigate to **Citywide** → **<new_page_name>**. In this example, the new page name is **Operator: Custom Operations** (see Figure 6-69).
Open the left and right panels by clicking the correspondent borders of the window.

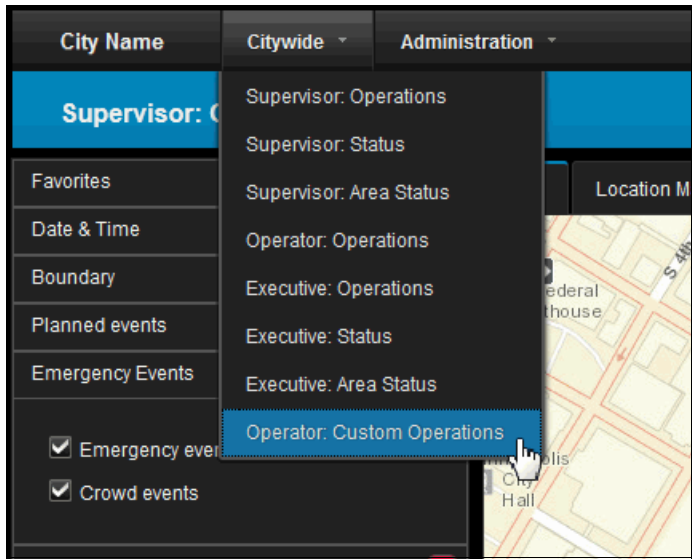


Figure 6-69 Selecting the new portal page

6.9.3 Change the city name on the top of the user interface

Change the city name to a custom string:

1. Navigate to **Administration** → **Portal Administration**.
2. Click **Portal User Interface** → **Manage Pages**.
3. Click **Content Root**.

4. For the City Name label, click the **Edit Page Properties** icon (see Figure 6-70).

The screenshot shows the 'Manage Pages' interface. At the top, there is a search bar with a dropdown menu set to 'Title starts with' and a 'Search' button. Below the search bar, there is a breadcrumb trail: 'Select Page > Content Root'. Underneath, there are buttons for 'New Page', 'New Label', and 'New Page from...'. The main part of the interface is a table with the following columns: 'Title', 'Unique name or Identifier', 'Status', and a set of action icons. The 'City Name' row is highlighted with a red box. The action icons for the 'City Name' row are also highlighted with a red box, and a tooltip labeled 'Edit Page Properties' is visible over the 'Edit' icon.

Title	Unique name or Identifier	Status	Actions
City Name	ioc.Home	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Home	ibm.portal.Home	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Administration	ibm.portal.Administration	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Hidden Pages	ibm.portal.HiddenPages	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Search Center	ibm.portal.Search	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Page Customizer	ibm.portal.Page Customizer	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Application Root	wps.application.root	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Tag Center	ibm.portal.Tagging	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]
Solution Administration	ioc.SolutionAdministration	Active	[Icons: Up, Down, Edit, Copy, Paste, Delete]

Figure 6-70 Editing page properties for the City Name label

5. Change the text in the Title field. For example, enter City A and click **OK** (see Figure 6-71).

Note: To provide different translations for different languages, expand the Advanced options section and click **I want to set titles and descriptions**. Find the required language and edit the corresponding text.

Edit label: City Name

Title:

Unique Name:

Note: If the unique name you entered for this page already exists, it will not be created or updated.

Friendly URL name:

Theme:

Icon:

Aggregation - Render Mode:
This setting will revert to SSA during runtime if the theme assigned with the page does not support CSA.

Inherit Parent Render Mode
 Client Side Aggregation - Rendering
 Server Side Aggregation - Rendering

Page Cache Options

Advanced options
This page supports (at least one must be selected):
 HTML

[I want to set titles and descriptions.](#)

[I want to set parameters.](#)

Control Display through Rule Mapping
Show or Hide page rule: No rule mapped

[I want to edit associations.](#)

Figure 6-71 Changing name of the City Name label

- Switch back to the user's view and notice that the text was changed in the upper-left corner of the user interface (see Figure 6-72).

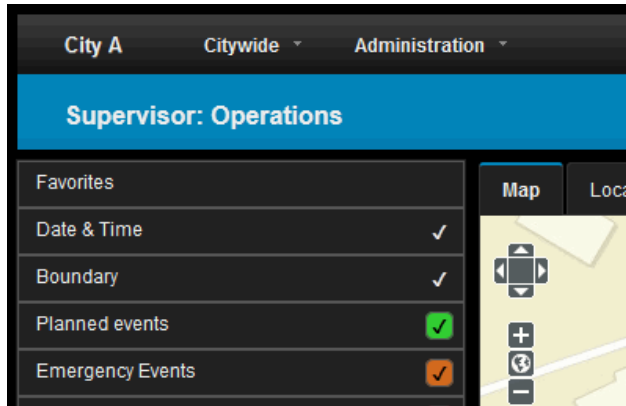


Figure 6-72 The changed title on the user interface

6.9.4 Customizing titles and IBM logos in IBM Intelligent Operations Center

For detailed information about customizing titles and IBM logos in IBM Intelligent Operations Center, see the technote *Customize titles and IBM logos in IBM Intelligent Operations Center*.

<http://www-01.ibm.com/support/docview.wss?uid=swg21656917>



Visualizing information with maps

This chapter describes the use of Geographic Information Systems (GIS) features in IBM Intelligent Operations Center. It describes the visualization of information with maps, providing situational awareness for various events and incidents.

The festival in City A continues medical incidents are reported, emergency crews are located, and medical attention is provided in time to ensure the smooth progress of the event.

This chapter includes the following topics:

- ▶ 7.1, “GIS terminology” on page 270
- ▶ 7.2, “Creating maps” on page 274
- ▶ 7.3, “Use case overview” on page 285
- ▶ 7.4, “Implementation summary” on page 286
- ▶ 7.5, “Implementation details” on page 286

7.1 GIS terminology

GIS is a relatively new suite of interrelated technologies. The terminology associated with GIS is not well known. This section provides a list of concepts that are required to understand the topics in this chapter. There are generic GIS concepts and concepts specific to IBM Intelligent Operations Center.

7.1.1 Generic GIS terminology

The following terms are important.

Geographic Information System (GIS)

GIS is a system designed to capture, store, manage, and present all types of geographic data. The data is commonly represented on maps. A GIS helps you to view and interpret data in relation to other data, and to take action.

Layer

A *layer* is the visual representation of a geographical data set in any digital map environment. Conceptually, a layer is a slice or stratum of the geographical reality in a particular area, and is more or less equivalent to a legend item on a paper map. On a roadmap, for example, roads, national parks, political boundaries, and rivers might be considered different layers.

Esri

Esri is the international supplier of GIS software, maps, and geodatabase management applications. See the Esri website:

<http://www.esri.com/>

OpenStreetMap

OpenStreet Map is an open initiative and collaborative project to create a free map of the world. The objective is the development and distribution of free geospatial data, such as street maps. See the OpenStreetMap website:

<http://www.openstreetmap.org>

OpenLayers¹

OpenLayers is an open source JavaScript library to load, display, and render maps from multiple sources on web pages.

XYZ layer (OpenLayers)

X and Y represent the column and row of the grid, and Z represents the zoom level.

Note: The IBM Intelligent Operations Center supports maps that comply with OpenLayer's XYZ layer and OpenStreetMap. IBM Intelligent Operations Center uses WGS84 standards for representing geospatial information.

Polygon¹

On a map, a *polygon* is a closed shape defined by a connected sequence of x,y coordinate pairs, where the first and last coordinate pair are the same and all other pairs are unique.

¹ From OpenLayers at <http://openlayers.org/>

Map tiles

Map tiles support the fast visualization of large data sets in a web map. Map tiles are square bitmap graphics of a specific size (typically 256 x 256 pixel images or 64 x 64 pixel images for mobile users). They are arranged as a grid to display a map.

Geodetic Control Networks²

Geodetic Control Networks consist of stable, identifiable points tied together by extremely accurate observations. From these observations, coordinates are computed and published to provide the common basis that is important to surveying and mapping activities.

WGS 84³

The World Geodetic System (WGS) is an earth-centered, earth-fixed terrestrial reference system and geodetic datum. It is based on a consistent set of constants and model parameters that describe the earth's size, shape, and gravity and geomagnetic fields. WGS 84 is the standard U.S. Department of Defense definition of a global reference system for geospatial information and is the reference system for the Global Positioning System (GPS). It is compatible with the International Terrestrial Reference System (ITRS).

7.1.2 GIS terminology specific to IBM Intelligent Operations Center

This section covers basic map-related terminology used in the IBM Intelligent Operations Center solution.

Base map

A *base map* is a map that depicts background reference information, such as landforms, roads, landmarks, and political boundaries, onto which other thematic information is placed. A base map is used for locational reference and often includes a geodetic control network as part of its structure.

In this book's scenario, the map of City A is the base map.

The IBM Intelligent Operations Center GIS map supports the following base map types:

- ▶ XYZ: OpenLayer's XYZ to support tiles (tiles of map data from a tile server)
- ▶ OpenStreetMap

A base map definition includes a name, default map indicator, map type, tile server URL, attribution, latitude and longitude, and zoom level.

Namespace

The IBM Intelligent Operations Center map setting namespace is `DefaultMapSettings_IOC`.

Map setting

A *map setting* is a collection of base maps defined for a namespace. It also contains the zoom level, center point, attribution, and information about whether it is the default map.

² From http://www.ngs.noaa.gov/FGCS/tech_pub/1984-stds-specs-geodetic-control-networks.htm

³ From World Geodetic System 1984 at http://www.oosa.unvienna.org/pdf/icg/2012/template/WGS_84.pdf

Location map

A *location map* is a map or plan containing interactive areas that have been defined in IBM Intelligent Operations Center. Hierarchical areas can be set up, and an area can appear on multiple maps. An event can be associated with one or more of these areas.

For example, a diagram of seating areas in a major sports stadium in City A can be defined so that events that have occurred can be associated with the appropriate area.

Collection

A *collection* is a group of location maps. A collection usually represents a location. Examples of collections are City stadium and Sheriff's office.

Category

A *category* contains collections. Categorizing collections enables the user to easily access collections. Examples of categories are Police stations, Stadiums, and Schools.

Area

An *area* is a user-defined polygon on the location map. Each area is assigned a name.

Content viewer

The *content viewer* is the extensible portlet that is predefined and available on the Operations pages. It displays the base map, the location map, and the list. The content viewer portlet shows the data items filtered by the settings specified in the filter panel. Figure 7-1 on page 273 shows the content viewer portlet and the filter panel.

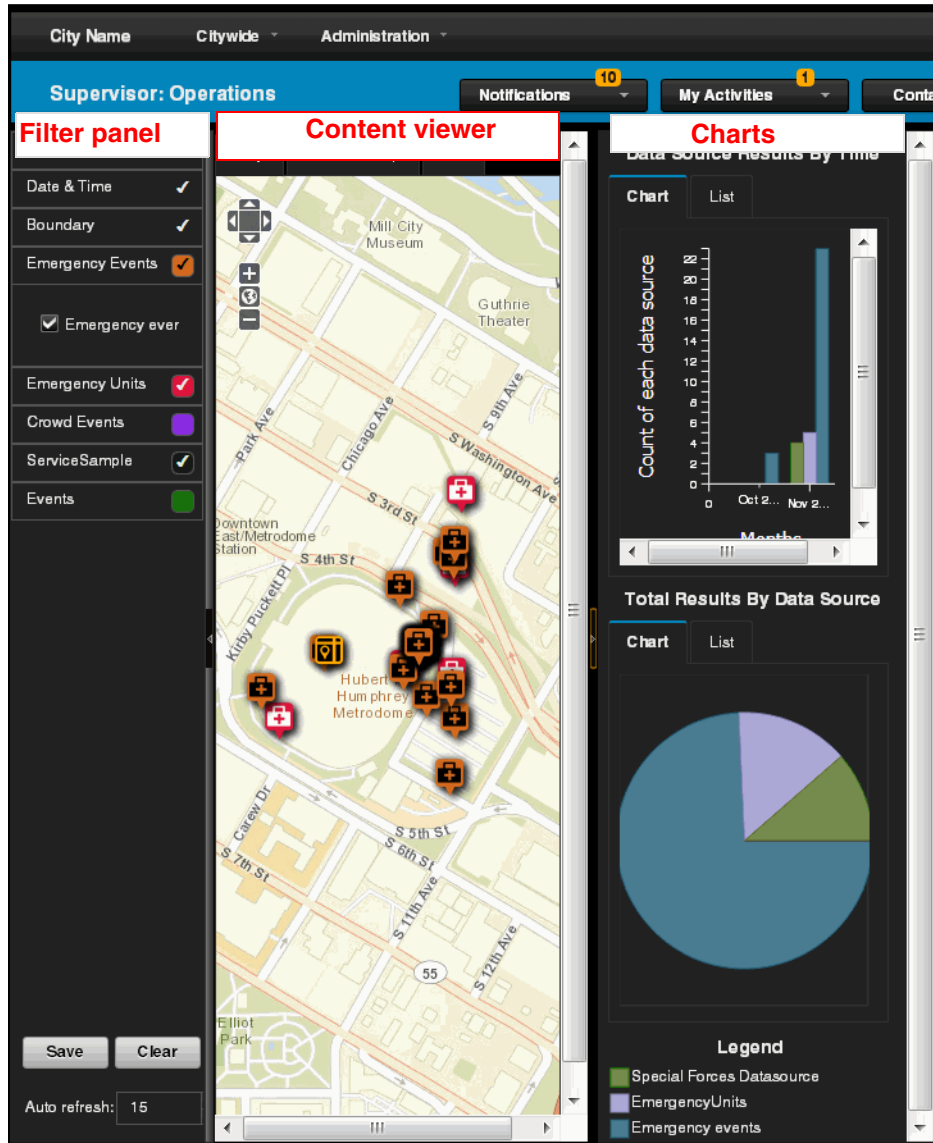


Figure 7-1 IBM Intelligent Operations Center content viewer portlet

Filter panel

In the appropriate operations view, use the filter panel to define what data is displayed in the content viewer and charts.

Bounding box

The *bounding box* is the perimeter of the map display. Only events that are in the bounding box view display on the map or list on the location map.

Map marker

A *map marker* is a general collective term for a marker on the map that indicates the presence and location of data items or a location map collection.

A map marker icon that is selected during data source creation is used to indicate the presence and location of an item on the map. The filter panel color is used for the map markers.

Preview card

The *preview card* displays minimal and key properties of the data record. It is launched by clicking the map marker. The full properties can be viewed by clicking the More details option.

Full property dialog

The full property dialog displays all details of the data source item when the user selects the More details link on the preview card. When the More Actions option is clicked on the preview card, other actions that can be taken on the data item are displayed. These include update, cancel, routing to a currently running standard operating procedure, association to a standard operating procedure, and custom actions that can be performed. These options depend on what actions were configured for this data source during data source creation.

7.2 Creating maps

The first step in using GIS capabilities in IBM Intelligent Operations Center is to create maps. This section explains how to create the two types of maps available in IBM Intelligent Operations Center:

- ▶ Base maps
- ▶ Location maps

7.2.1 Creating a base map

Create a base map on IBM Intelligent Operations Center:

1. Log in as the solution administrator.
2. Click **Administration** → **Solution Administration**.
3. Click **Configuration Tools** → **Geospatial Map**.
4. Click **Create** on the Base Maps tab to create a new base map.
5. In the Base Map Properties window, follow these steps:
 - a. Enter a name for the map.
 - b. Select a map type from the list of base map providers. Choose one of the two options in the drop-down list:
 - XYZ
 - OpenStreetMap URL

Depending on the provider you choose, you might need to complete additional fields.

Figure 7-2 on page 275 shows the Base Map Properties for this use case.

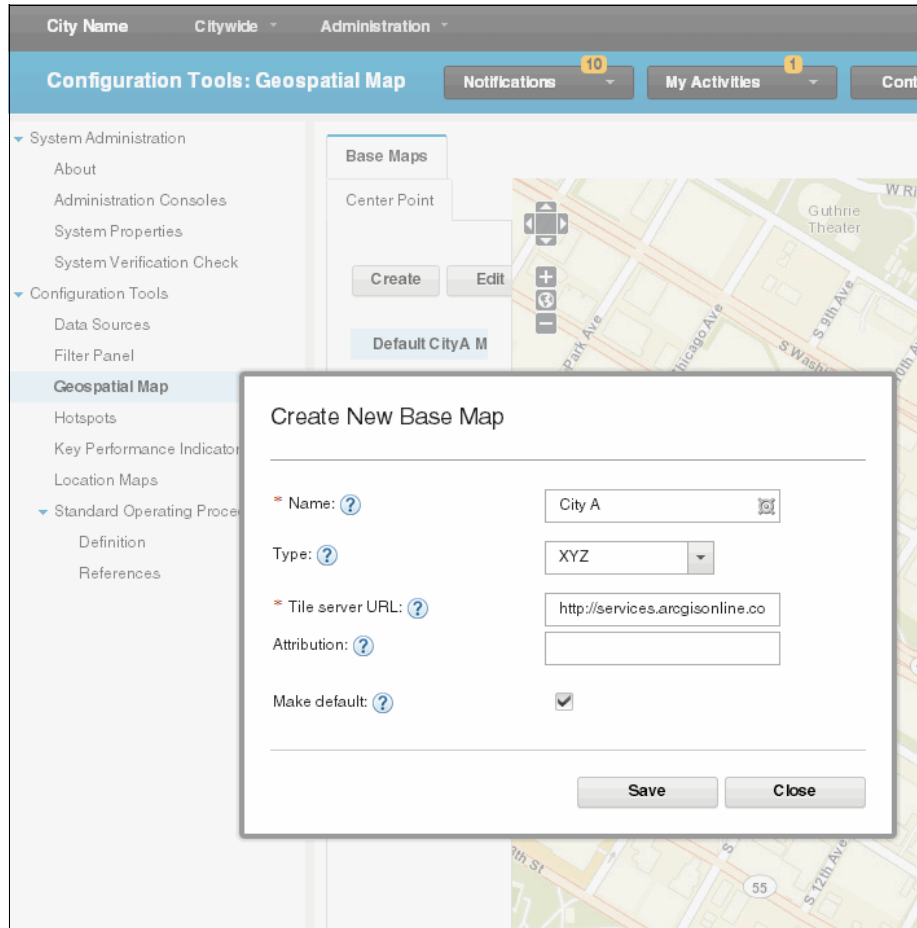


Figure 7-2 Base map creation

6. Click **Save**.

7.2.2 Creating location maps

This section describes the creation of location maps, both from the solution user interface (UI) and programmatically with the Location Map Representational State Transfer (REST) service.

Uploading location map images

You can create a map that is displayed on the Location Map tab. As a precursor to that task, you can upload map images. Any images that you upload are then available to use when you create a new location map.

Follow these steps:

1. Log in as a solution administrator.
2. Click **Administration** → **Solution Administration** → **Configuration Tools** → **Location Maps**.
3. Click **Location Map images** → **Upload**.
4. Browse to the image file on your local file system, and click **Open** to upload it to the server.

The image is now available on the Location Maps list.

The images can be deleted only by an administrator, directly on the application server in the following directory:

```
/opt/IBM/WebSphere/wp_profile/installedApps/cell1/ioc_theme_ear.ear/ioc_theme_stat  
ic_js_web.war/images/location_maps
```

Note: The accepted formats for location map images are PNG, JPG, GIF, and SVG.

Creating a location map using the location maps configuration tool

Administrators can create, update, or delete a location map in the solution. To create a location map in the solution, use the location maps configuration tool as described in the following steps:

1. Log in as a solution administrator.
2. Click **Administration** → **Solution Administration** → **Configuration Tools** → **Location Maps**.
3. Click **Create** → **Browse** to locate a previously uploaded image on the server.
4. In the Map title field, enter a name for the map and click **Next**.
5. Choose your map collection and category:
 - a. Click **Choose Existing Category**, and in the Category list, select an existing category for your location map. Alternatively, click **Create New Category**, and in the Category Name field, enter the name of a new category.
 - b. Click **Choose Existing Collection**, and in the Collection list, select an existing collection for your location map. Alternatively, select an existing **Category** and click **Create New Collection**, enter the name of a new collection, and enter its latitude and longitude coordinates.
6. Optionally, click **Next** to create the named areas for your location map.
Enter an area name and area description, and click **Save**.

Figure 7-3 on page 277 shows the creation of a location map.

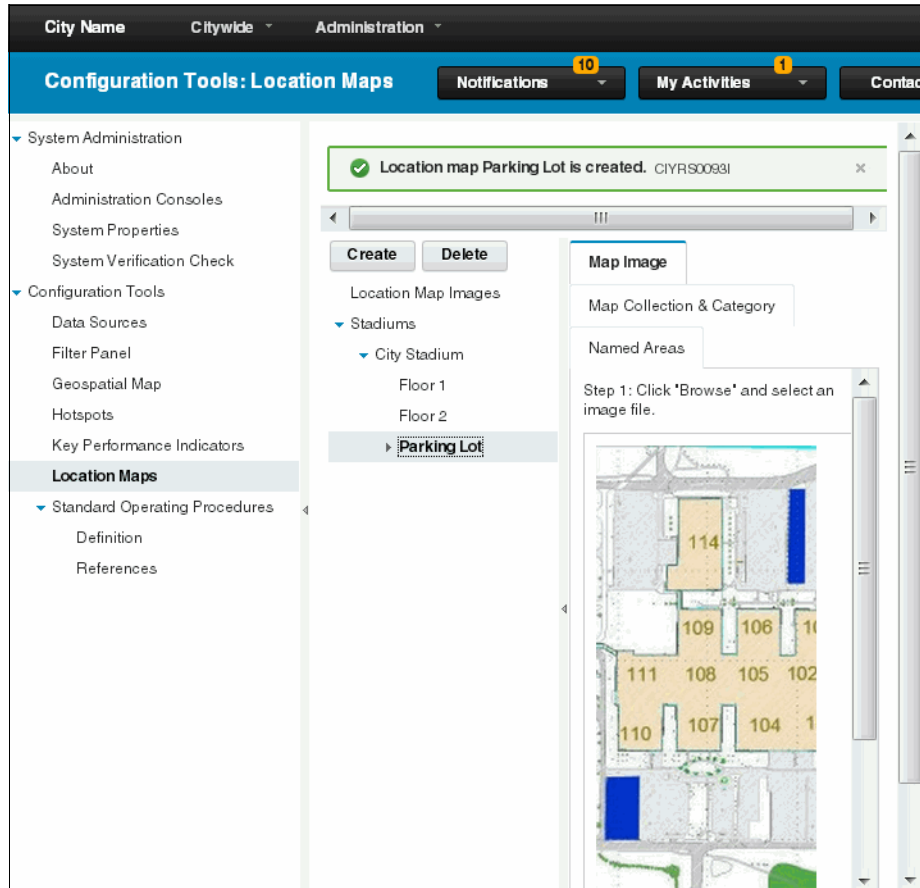


Figure 7-3 Creating a location map

Creating location maps with the location map REST service

The location map service provides functionality to create, read, update, and delete location map categories, collections, and location maps. It also provides functionality to create, read, and delete area locations.

Note: This section illustrates the creation of a location map with the location map REST service interface. A server-side Java interface is also available; see *Java API documentation* at this website:

<http://bit.ly/1hU0Md9>

Create a location map using the location map REST service:

1. Use the GET method to query the categories to check whether your category of choice already exists.

Request: `/ibm/ioc/api/location-map-service/categories`

Java service: `com.ibm.ioc.locationmap.CategoryDataService`

Example 7-1 on page 278 shows the response.

Example 7-1 Query category response

```
[1]
0: {
  id: "CATEGORY_1"
  label: "{\"group\":\"LocationMap\",\"key\":\"sample_data_stadiums\"}"
  i18nLabel: "Stadiums"
  lastUpdateDate: 1375544499883
  messages: [0]
}
```

2. If the category that you need does not already exist, use the POST method to create the category before you add the collection and map as shown in Example 7-2.

Request: `/ibm/ioc/api/location-map-service/categories`

Java service: `com.ibm.ioc.locationmap.CategoryDataService`

Example 7-2 POST request payload to create Category Schools

```
{"label": "{\"group\":\"LocationMap\",\"key\":\"Schools\"}"}
```

Example 7-3 shows the response. The response shown is from a REST client (used for documentation purposes).

Example 7-3 Response to POST (creation of new Category Schools)

```
{
  id: "d4e732cf-24dd-4663-9245-324d3dbf6475"
  label: "{\"group\":\"LocationMap\",\"key\":\"Schools\"}"
  i18nLabel: "Schools"
  lastUpdateDate: null
  messages: [1]
  0: {
    messageId: "CIYRS0089I"
    messageText: "{\"key\":\"CIYRS0089I\",\"group\":\"LocationMapLobApp\"}"
    i18nMessageText: "Category Schools is created."
  }-
-
}
```

Note: You need to authenticate as an administrative user to create resources, such as categories, collections, and location maps.

3. Use the GET method to query the collections to check whether the collection to which you want to add the location map already exists.

Request: `/ibm/ioc/api/location-map-service/collections`

Java Service: `com.ibm.ioc.locationmap.CollectionDataService`

Example 7-4 shows the response.

Example 7-4 Query collections response

```
[1]
0: {
  id: "COLLECTION_1"
  label: "{\"group\":\"LocationMap\",\"key\":\"sample_data_city_stadium\"}"
  i18nLabel: "City Stadium"
}
```



```
location: "POINT (-93.258269 44.974119)"
categoryId: "CATEGORY_1"
lastUpdateDate: 1382340623039
messages: [0]
}
```

4. If the collection you need does not already exist, use the GET method to query all categories as shown in Example 7-5.

Request: `/ibm/ioc/api/location-map-service/categories`

Java service: `com.ibm.ioc.locationmap.CategoryDataService`

Example 7-5 Query categories response

```
[2]
0: {
id: "d4e732cf-24dd-4663-9245-324d3dbf6475"
label: "{\"group\":\"LocationMap\",\"key\":\"Schools\"}"
i18nLabel: "Schools"
lastUpdateDate: 1385581265439
messages: [0]
}-
1: {
id: "CATEGORY_1"
label: "{\"group\":\"LocationMap\",\"key\":\"sample_data_stadiums\"}"
i18nLabel: "Stadiums"
lastUpdateDate: 1375544499883
messages: [0]
}
```

5. With the categoryId of the category for which you want to create the collection, (CATEGORY_1 in this example), use the POST method to create the collection to which you want to add the location map as shown in Example 7-6.

Request: `/ibm/ioc/api/location-map-service/collections`

Java Service: `com.ibm.ioc.locationmap.CollectionDataService`

Example 7-6 POST request payload

```
{
"label": "{\"group\":\"LocationMap\",\"key\":\"Art Festival Venue\"}",
"i18nLabel": "Art Festival Venue",
"location": "POINT (-93.268269 44.974119)",
"categoryId": "CATEGORY_1"}
}
```

6. The response is shown in Example 7-7.

Example 7-7 Response to POST (creation of new collection)

```
{
id: "be682186-00c7-42de-bc79-7c98946ccf32"
label: "{\"group\":\"LocationMap\",\"key\":\"Art Festival Venue\"}"
i18nLabel: "Art Festival Venue"
location: "POINT (-93.268269 44.974119)"
categoryId: "CATEGORY_1"
lastUpdateDate: null
messages: [1]
0: {
```

```
messageId: "CIYRS0091I"
messageText: "{\"key\":\"CIYRS0091I\",\"group\":\"LocationMapLobApp\"}"
i18nMessageText: "Collection Art Festival Venue is created."
}-
-
}
```

7. Query the collections again as shown in Example 7-8.

Request: `/ibm/ioc/api/location-map-service/collections`

Java Service: `com.ibm.ioc.locationmap.CollectionDataService`

Example 7-8 Query Collections

```
[2]
0: {
id: "be682186-00c7-42de-bc79-7c98946ccf32"
label: "{\"group\":\"LocationMap\",\"key\":\"Art Festival Venue\"}"
i18nLabel: "Art Festival Venue"
location: "POINT (-93.268269 44.974119)"
categoryId: "CATEGORY_1"
lastUpdateDate: 1385583591646
messages: [0]
}-
1: {
id: "COLLECTION_1"
label: "{\"group\":\"LocationMap\",\"key\":\"sample_data_city_stadium\"}"
i18nLabel: "sample_data_city_stadium"
location: "POINT (-93.258269 44.974119)"
categoryId: "CATEGORY_1"
lastUpdateDate: 1385583615599
messages: [0]
}
```

8. Use the GET method to query location maps to get the list of currently available location maps.

Request: `/ibm/ioc/api/location-map-service/maps`

Java Service: `com.ibm.ioc.locationmap.MapDataService`

Example 7-9 shows the response.

Example 7-9 Query maps response

```
[3]
0: {
id: "2414542a-ec36-463b-8792-b344468392e1"
collectionId: "COLLECTION_1"
label:
"{\"group\":\"LocationMap\",\"key\":\"i18n_ff54be46-e3fb-44a7-8a87-1ad232f404ba\"}"
i18nLabel: "Floor 1"
backgroundImage:
"/ibm/ioc/theme/static/js/images/location_maps/CityStadium.jpg"
backgroundImageHeight: "620"
backgroundImageWidth: "723"
lastUpdateDate: 1384815491560
messages: [0]
}-
```

```

1: {
  id: "81b5877e-f6ac-43b6-80f1-5966c24fc92a"
  collectionId: "COLLECTION_1"
  label:
    "{"group":"LocationMap","key":"i18n_80d45e17-625b-42ed-8045-4dbbf11ff477"}"
  i18nLabel: "Floor 2"
  backgroundImage: "/ibm/ioc/theme/static/js/images/location_maps/nfl_min.png"
  backgroundImageHeight: "525"
  backgroundImageWidth: "400"
  lastUpdateDate: 1384815922999
  messages: [0]
}-
2: {
  id: "46003827-7c1b-49dc-a20b-78ccf0230905"
  collectionId: "COLLECTION_1"
  label:
    "{"group":"LocationMap","key":"i18n_53c4d3f5-1dd6-4c35-8ed9-3cff58c8b1a1"}"
  i18nLabel: "Parking Lot"
  backgroundImage:
    "/ibm/ioc/theme/static/js/images/location_maps/parkinglotmap.jpg"
  backgroundImageHeight: "268"
  backgroundImageWidth: "530"
  lastUpdateDate: 1385070287865
  messages: [0]
}

```

9. Use the POST method to create the location map with the `collectionId` of the collection as queried in step 7 on page 280, `COLLECTION_1` in this example. Example 7-10 shows the POST payload.

Request: `/ibm/ioc/api/location-map-service/maps`

Java Service: `com.ibm.ioc.locationmap.MapDataService`

Example 7-10 POST request payload

```

{
  "backgroundImage":"/ibm/ioc/theme/static/js/images/location_maps/Police_Station.jpg",
  "backgroundImageWidth":723,
  "backgroundImageHeight":620,
  "collectionId":"COLLECTION_1",
  "label":{"\group\":"LocationMap","\key\":"PoliceStation\"} }

```

10. The response is shown in Example 7-11.

Example 7-11 POST response (create new location map)

```

{
  id: "58b19281-b6af-4ad3-85c6-15b4a0a08e53"
  collectionId: "COLLECTION_1"
  label: "{"group":"LocationMap","key":"PoliceStation"}"
  i18nLabel: "PoliceStation"
  backgroundImage:
    "/ibm/ioc/theme/static/js/images/location_maps/Police_Station.jpg"
  backgroundImageHeight: "620"
  backgroundImageWidth: "723"
}

```

```

lastUpdateDate: null
messages: [1]
0: {
messageId: "CIYRS0093I"
messageText: "{\"key\":\"CIYRS0093I\",\"group\":\"LocationMapLobApp\"}"
i18nMessageText: "Location map PoliceStation is created."
}-
-
}

```

11. Query all defined area locations for a specific location map.

Request: **/ibm/ioc/api/location-map-service/areaLocations**

Java Service: **com.ibm.ioc.locationmap.AreaLocationDataService**

Example 7-12 shows the response.

Example 7-12 Query area locations response

```

[2]
0: {
id: "Home team VIP Seating"
name: "Home team VIP Seating"
location: "POLYGON ((227 101, 141 102, 143 72, 240 70, 241 70, 227 101))"
labelDescription:
"{\"group\":\"LocationMap\",\"key\":\"i18n_0527a250-61f5-4342-8873-c96d9170f7c0\"}"
i18nLabelDescription: "VIP area home team side"
lastUpdateDate: 1384874161141
parentAreaId: null
children: [0]
messages: [0]
mapId: "2414542a-ec36-463b-8792-b344468392e1"
}-
1: {
id: "Visitor VIP Seating"
name: "Visitor VIP Seating"
location: "POLYGON ((186 227, 138 229, 138 201, 188 200, 186 227))"
labelDescription:
"{\"group\":\"LocationMap\",\"key\":\"i18n_568efa3c-7e60-4116-8d9a-eb145e24aaff\"}"
i18nLabelDescription: "Visitor Team Side VIP Seating"
lastUpdateDate: 1384874161141
parentAreaId: null
children: [0]
messages: [0]
mapId: "2414542a-ec36-463b-8792-b344468392e1"
}

```

12. If needed, use the POST method to create an area location.

7.2.3 Configuring map settings

The map settings resource is used to get, create, update, and delete map settings resources for a namespace.

For example, the following query is for the `namespace` parameter equal to `DefaultMapSettings_IOC`.

Request:

```
/ibm/ioc/api/mapsettings-service/mapsettings?namespace=DefaultMapSettings_IOC
```

Java service: `com.ibm.ioc.map.MapSettingsService`

Example 7-13 shows the sample response.

Example 7-13 Query MapSettingsService response

```
{
  baseMaps: [1]
  0: {
    name: {
      group: "MapSettings"
      locale: null
      key: "pre_populated_basemap_label"
      value: null
      id: null
      lastUpdateDate: null
      messages: [0]
    }-
    defaultMap: true
    url:
      "http://services.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer/
      tile/{z}/{y}/{x}"
    mapType: "XYZ"
    googleMapType: "STREET"
    attribution: ""
  }-
  -
  location: {
    lon: "-93.2570815086443"
    lat: "44.972980544218"
    zoom: 16
  }-
  namespace: "DefaultMapSettings_IOC"
  id: 35
  lastUpdateDate: 1382937761323
  messages: [0]
}
```

7.2.4 Map extension points

IBM Intelligent Operations Center provides map extension points that allow developers to integrate with the map views and control the map views. The extension points are available as Dojo topics.

Note: The Dojo Topic System, also known as Publish and Subscribe, is a simple way for ambiguous objects, widgets, or code to communicate with other widgets and objects without prior knowledge of each other. See the *Publish/Subscribe* section in the *Events with Dojo* tutorial at this website:

<http://dojotoolkit.org/documentation/tutorials/1.7/events/>

Using Dojo topics, developers can create custom code for IBM Intelligent Operations Center that integrates with and controls existing IBM Intelligent Operations Center portlets. For example, a developer can write custom JavaScript code to programmatically make the content viewer portlet switch to a map and highlight a certain feature on the map.

Each available map topic is identified by a unique topic name that developers use when publishing events to this topic. For example, the topic used to highlight a feature on the map has the topic name of `/ibm/ioc/map/highlightMapFeature`.

Table 7-1 lists the Dojo topics provided by IBM Intelligent Operations Center that developers can use as extension points. The example column shows the single line of code that developers need to write in JavaScript to be able to use the extension point.

Table 7-1 Map extension points

Topic name	Description	Example
<code>/ibm/ioc/map/highlightMapFeature</code>	Publishing to this topic opens the map tab in the content viewer and selects the features with the IDs specified.	<code>topic.publish("/ibm/ioc/map/highlightMapFeature", layerId, dataRecordIdArray);</code>
<code>/ibm/ioc/featureSelected</code>	When a feature is selected on the GIS map, it publishes to this topic. Code can listen to this topic and implement custom code when features are selected.	<code>topic.subscribe("/ibm/ioc/featureSelected", function(layerId, dataRecordIdArray) { });</code>
<code>/ibm/ioc/featureUnSelected</code>	Similar to the <code>featureSelected</code> topic, when a feature is unselected on the GIS map, it publishes to this topic.	<code>topic.subscribe("/ibm/ioc/featureUnSelected", function(layerId, dataRecordIdArray) { });</code>
<code>/ibm/ioc/locationmap/menu/select</code>	Publishing to this topic selects the location map with ID in the location map menu, causing it to display on the right pane.	<code>topic.publish("/ibm/ioc/locationmap/menu/select", id);</code>
<code>/ibm/ioc/map/openLocationMap</code>	Show the Location Map tab in the content viewer, and select and open the location map with the ID.	<code>topic.publish("/ibm/ioc/map/openLocationMap", id);</code>

7.3 Use case overview

A celebrity is spotted, and a crowd rapidly gathers. In the chaos, a man falls down, cuts his foot, and bleeds profusely. The festival operator creates a new event for the incident on the map and uses IBM Intelligent Operations Center to find the location of the nearest emergency unit and directs them to the location of the incident. The implementation steps described in this section are performed by the IBM Intelligent Operations Center administrator in preparation for the event to make sure that the operators will be able to handle the emergencies as they occur.

The data item information entered by the operator automatically sends the emergency event information to the medical unit system, which helps them to arrive prepared to treat the injured person.

The use case described in 9.4, “Routing data to the integration topic” on page 331 provides information that is complementary to this use case.

7.3.1 Actors

This use case involves the following actors:

- ▶ Festival operator: The person using IBM Intelligent Operations Center at the event operations center
- ▶ Emergency units: An ambulance on the event ground that can be dispatched to first aid events to provide assistance

7.3.2 Desired outcome

This proposed solution has the following objectives:

- ▶ Enable the event operators to find the emergency units that are close to the emergency event
- ▶ Dispatch the appropriate first aid emergency units quickly to provide the necessary assistance so that the festival continues without disruption

7.3.3 IBM Intelligent Operations Center programming artifacts

To implement the solution proposed in this use case, the following IBM Intelligent Operations Center REST services are used:

- ▶ Location map service
- ▶ Map settings service
- ▶ Spatial service

This solution also uses a data source with routing expression settings.

7.4 Implementation summary

The following summary shows the steps to implement this use case:

- ▶ Add data sources. The data sources for this use case are the Emergency Events data source and the Emergency Units data source.
- ▶ Create an emergency event when a first aid event occurs.
- ▶ Locate emergency units in the area and dispatch them.

7.5 Implementation details

This section includes step-by-step instructions to implement this use case.

7.5.1 Adding data sources

Two data sources are needed for this scenario:

- ▶ Emergency Events data source
- ▶ Emergency Units data source

In this scenario, the data for the two data sources is uploaded from comma-separated value (CSV) files. The CSV files are supplied with the samples under the directory `\SAMPLES\8201Data\sample_data_for_datasources`. See Appendix B, “Samples provided with this book” on page 383.

Follow the steps in 3.2.4, “CSV data source” on page 55 to create the data sources and upload the data. The two data sources have minimal data attributes, such as the start date, end date, last update time stamp, name, and location.

Figure 7-4 on page 287 shows the Supervisor:Operations view after creating the Emergency Events and Emergency Units data sources.

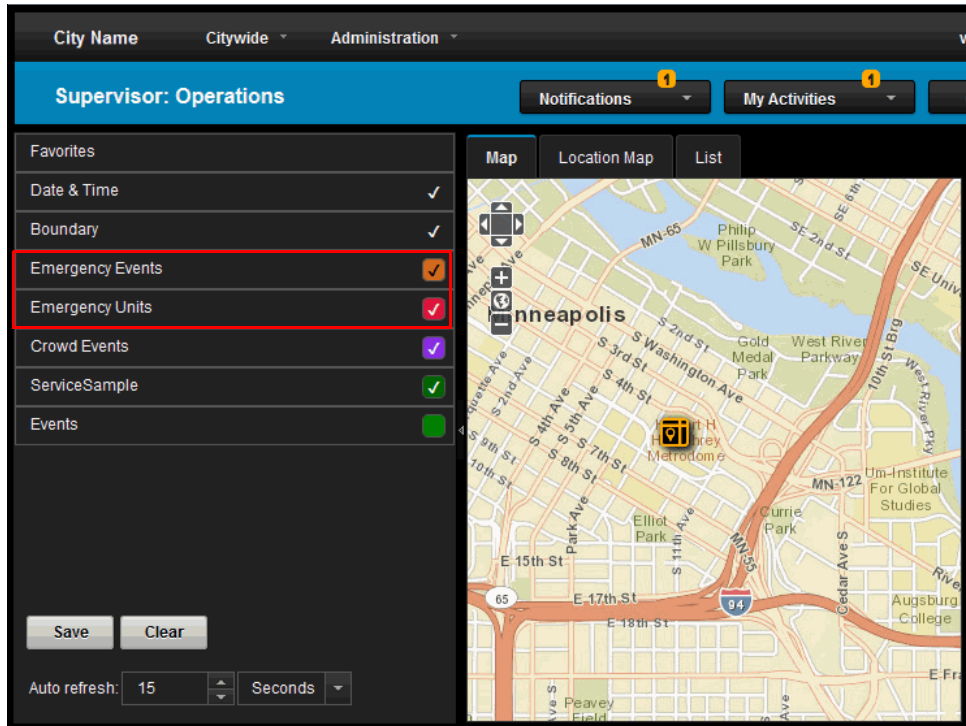


Figure 7-4 Adding the Emergency Events and Emergency Units data sources

A GIS layer is automatically added when a data source is configured in IBM Intelligent Operations Center. To add a GIS layer programmatically, create a data source, associated filter, and filter pane invoking the appropriate REST service.

7.5.2 Creating emergency events

You can create an emergency event in one of two ways:

- ▶ Manually, using the solution UI
- ▶ Programmatically, using the REST services

Using the solution UI

Create a new emergency event:

1. Log in to IBM Intelligent Operations Center as a city operator.
2. Click **Citywide** → **Supervisor:Operations**.
3. Click **More Actions** → **New item**.
4. Drag the **Emergency Events** new item and drop it on the map location where the emergency occurred (see Figure 7-5 on page 288).

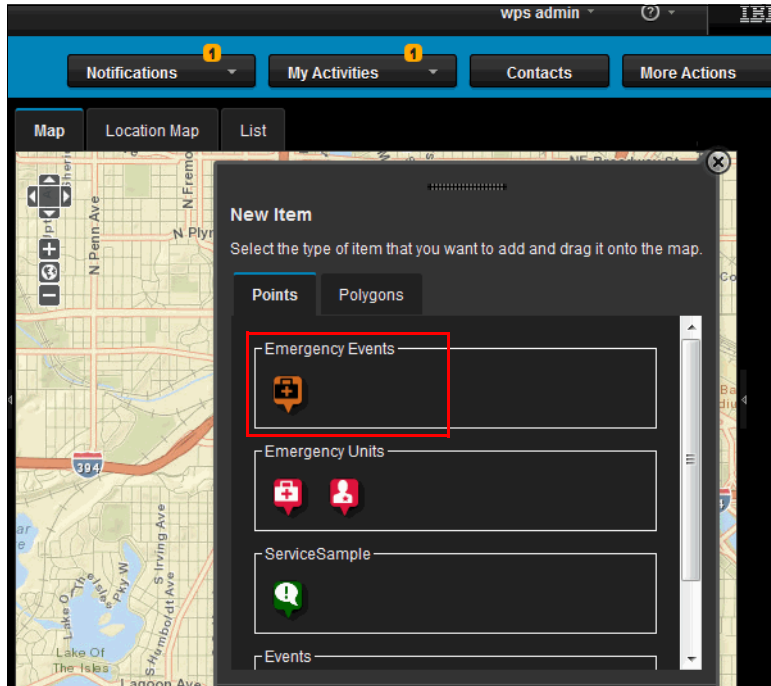


Figure 7-5 Adding an emergency event to the map

Note: Only users with write access to the data sources can drag items to the map.

5. Enter the emergency event information as shown in Figure 7-6 on page 289 and click **Save**.

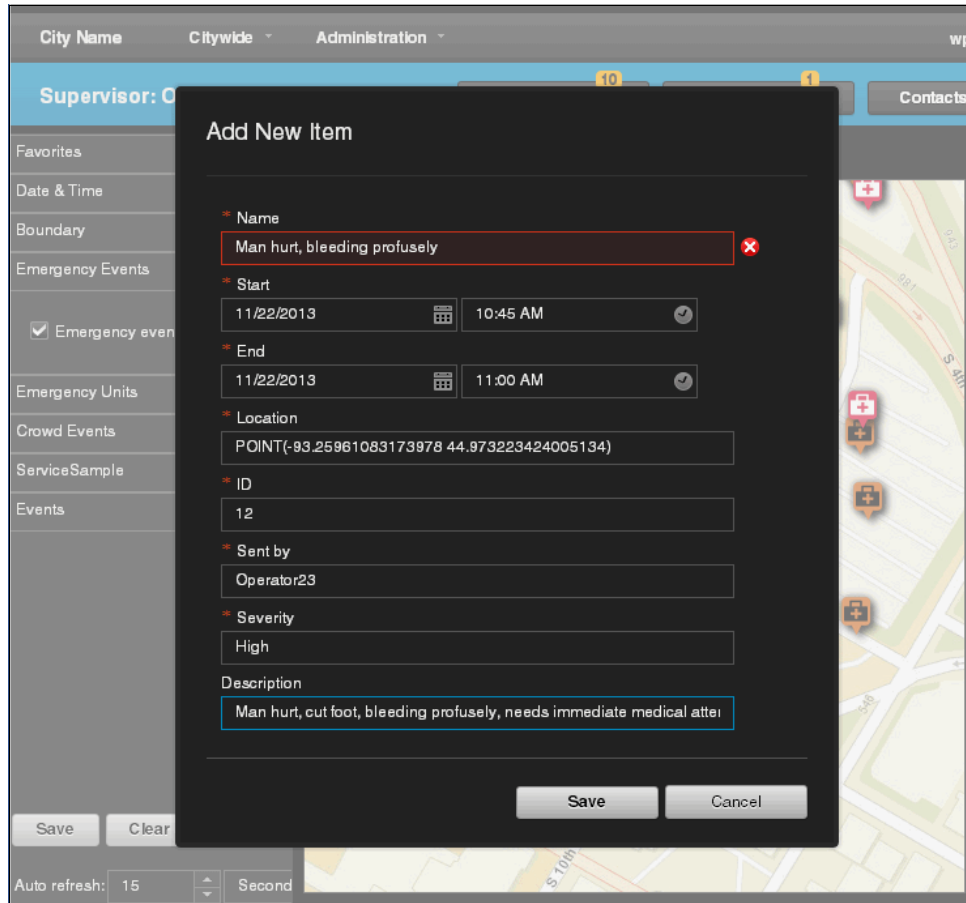


Figure 7-6 Creating an emergency event

Using REST services

An efficient event operation tool needs to allow emergency events to be reported from both the field and the operation center. Using the data injection service, a new emergency event can be added to the data emergency data source.

To add a new emergency event to the emergency events data source, an HTTP request is made to the data injection service with the POST method. The body of the request contains the new emergency event details. Example 7-14 shows the injection service URL to call.

Example 7-14 Injection service URL

```
https://<ioc_ibs>/ibm/ioc/api/data-injection-service/datablocks/<DATA_SOURCE_ID>/dataitem
```

Getting the data source Id is described in detail in “Identify data sources” on page 294.

Example 7-15 provides a sample content body to add a new emergency event.

Example 7-15 Request body

```
{
  "ASSESSMENTDATETIME": "2013-10-29T23:30:00-05:00",
  "DATASOURCEID": 19,
  "DELETEFLAG": false,
  "ENDDATETIME": "2013-11-06T22:30:00-05:00",
```

```
"LASTUPDATEDATETIME": "2013-11-18T16:10:10-05:00",
"LOCATION": "POINT(-93.25676500799601 44.97392549261744) ",
"NAME": "Man Down3",
"PROPERTY_2": "2",
"PROPERTY_3": "admin@AA.com",
"PROPERTY_4": "High",
"STARTDATETIME": "2013-10-29T23:30:00-05:00",
"TIMEZONEOFFSET": -300,
"lastUpdateTime": 1384827010476
}
```

If the injection is successful, the HTTP response has a status code of 200, and a copy of the newly created event is added to the HTTP response body. Take note of the record ID returned in the response because it might be needed later.

7.5.3 Finding nearby medical units

There are two ways the solution can be implemented to help operators find nearby medical units:

- ▶ Using the correlation topic
- ▶ Using the spatial service

The correlation topic is used for correlation between items from different data sources. The configuration can be performed by a business user in production to correlate different events. This correlation is displayed as a notification from the system.

When a custom user interface needs to be built to show the correlation, or if the correlation needs to drive other processes or invoke services, the IBM Intelligent Operations Center developer can provide these extensions by querying the spatial service to obtain event location details and then process the results, as needed.

Using the correlation topic

Follow these steps during IBM Intelligent Operations Center configuration for the event:

1. Configure system properties for correlation:
 - a. Click **Administration** → **Solution Administration** → **System Properties**.
 - b. Update the properties shown in Table 7-2 on page 291 and click **Save**.

Table 7-2 Correlation settings in system properties

Property name	Description
radius	The maximum distance between the routed data item and correlated data items. The default value is 200.
distanceUnit	The unit in which the distance between data items is measured, which can be either meters, kilometers, or statute miles. The default value is METER. The other valid values are KILOMETER and STATUTE_MILE.
minutesBefore, minutesAfter	The maximum time difference between the routed data item and correlated data items. The time difference is measured in minutes and is relative to the start date of each data item. The default value is 120 for both minutesBefore and minutesAfter.
notificationSummaryThreshold	If the number of correlations for a particular data item exceeds the notification summary threshold, a summary notification is sent instead of individual notifications. The default value is 10.
targetGroups	The user groups to whom correlation notifications are sent. The default value is ["CityWideOperator", "CityWideSupervisor", "wpsadmins"].

- On the Basics tab of the data source configuration tool, set Data for correlation to **Yes** for the Emergency Units data source and click **Save**, as shown in Figure 7-7.

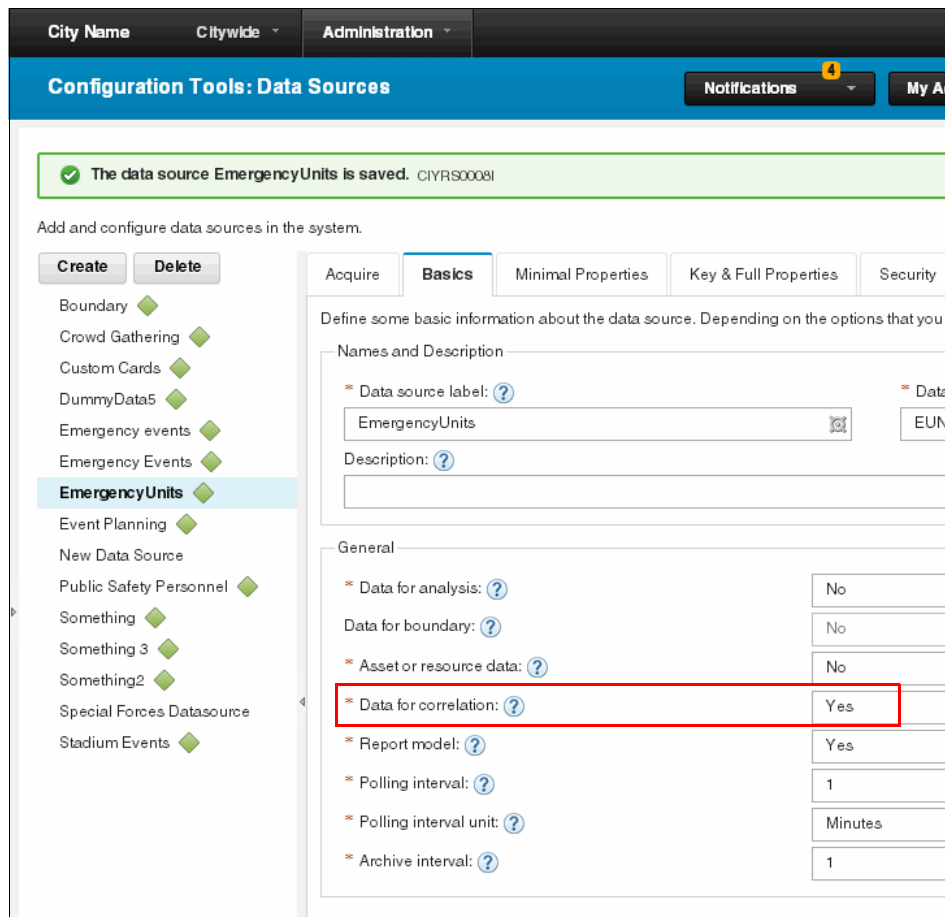


Figure 7-7 Configuring data source for correlation (Basics tab)

- On the Routing tab of the data source tool for the Emergency events data source, click **Add Routing Expression**. In Figure 7-8, select **SEVERITY** for the Property, **Equals** for the Operation, and **High** for the Criteria fields. For the Route to field, select **Correlation Topic**.

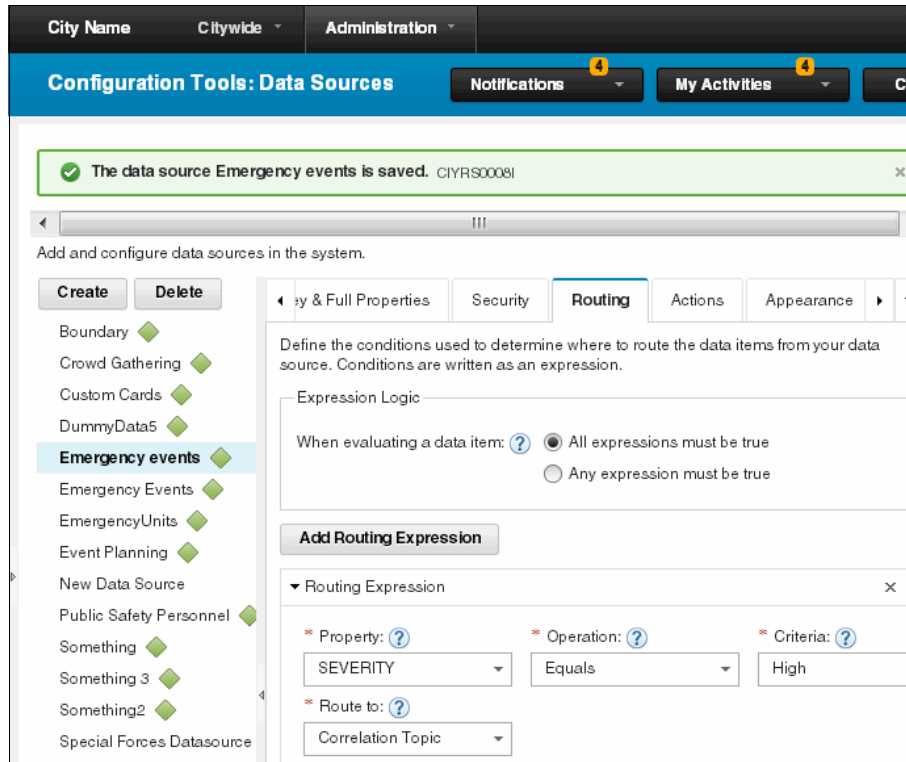


Figure 7-8 Configuring routing expression for correlation in Emergency events data source

The correlation process includes the following steps:

- The correlation topic reads the correlation settings from the system properties table.
- Within data sources that are activated for correlation, the correlation topic searches for data items that match all of the following conditions:
 - The data items are within the predefined spatial distance for correlation.
 - The data items occur within the predefined time difference for correlation.

The conditions are based on the correlation settings and are assessed relative to the routed data item.

- If the correlation topic finds correlations, it either sends individual correlation notifications, or a summary notification. The correlation topic sends a summary notification if the notification summary threshold is exceeded. The notifications are sent to the groups that are defined in the target groups system property.

Note: The distance and time values that are included in notifications are the maximum distance and time values that are defined in the correlation settings. Therefore, the values do not represent the exact distance and time between correlated data items.

In this example, *High* severity emergency events are routed to a correlation topic to be available for correlation with emergency units located in the same geographic radius and in the same time frame.

Operators receive notifications when a correlation is determined between the emergency event and the location of the emergency unit, within the specified radius, and at the same time as shown in Figure 7-9.

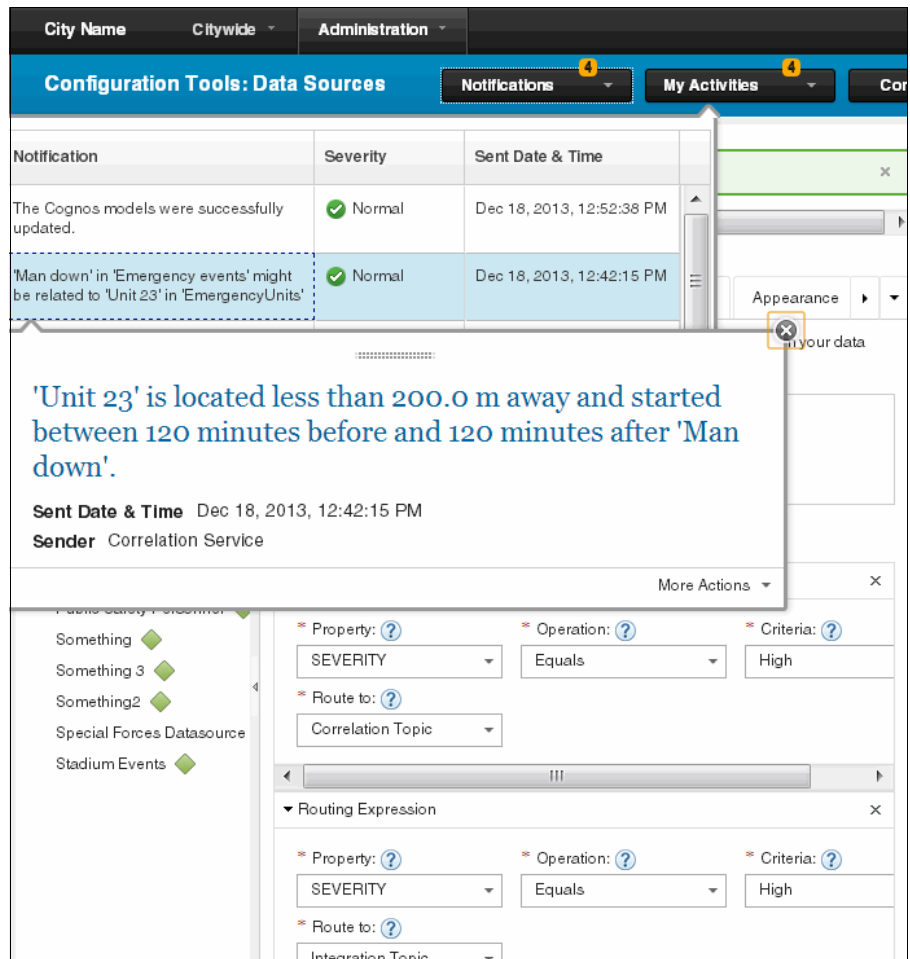


Figure 7-9 Notification of correlation between Emergency Event and location of Emergency Unit

When the operator clicks the notification, a message similar to the one shown in Example 7-16 is displayed.

Example 7-16 Notification message

'Unit 23' is located less than 200.0 m away and started between 120 minutes before and 120 minutes after 'Man down'.

The operator can then use the map to find the ambulance or medical unit as shown in Figure 7-10 on page 294.

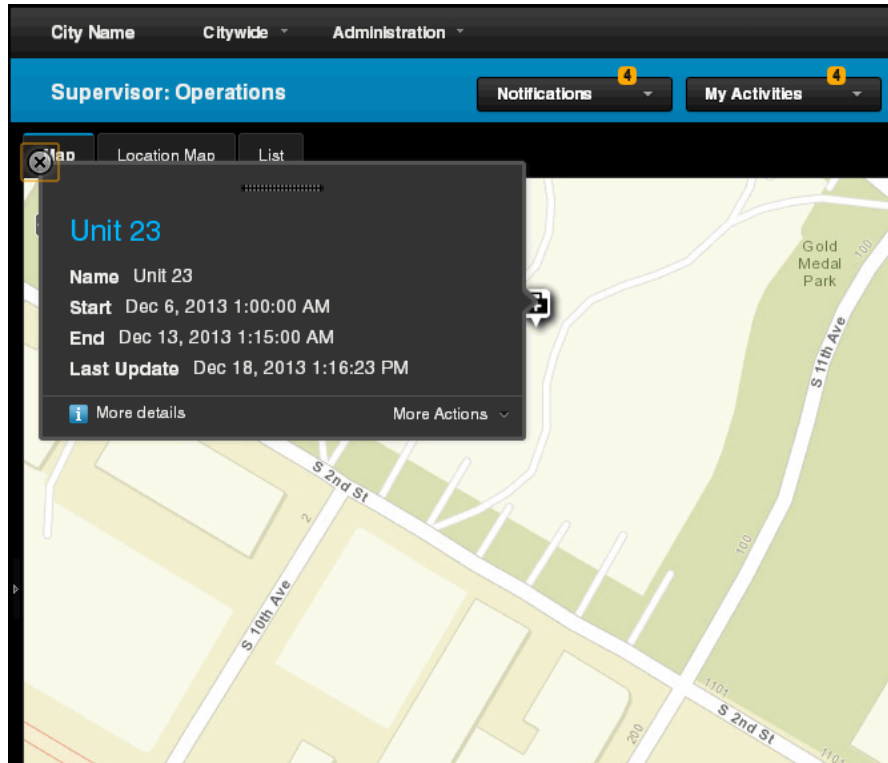


Figure 7-10 Medical unit's location on map

Using the spatial service

Another way to find nearby medical units is by using the spatial service that is described in this section. Here, the spatial service from the IBM Intelligent Operation Center REST services is used to locate the emergency units nearest to the first aid event coordinates. The spatial service provides access to the spatial-temporal data records stored in IBM Intelligent Operations Center as a result of creating a data source definition. The records provided by this service are displayed on the map.

The following steps summarize the implementation:

1. Identify the two data sources needed for first aid events and for emergency unit locations.
2. Get the location of the first aid event needing assistance.
3. Query the spatial service to get emergency units inside this polygon.

Identify data sources

In this scenario, the Emergency Events and Emergency Units data sources created in 7.5.1, "Adding data sources" on page 286 are used.

The data source IDs for both data sources are needed and can be obtained by querying the data source service to get a list of data sources and their IDs. The REST service URL shown in Example 7-17 needs to be called using the correct authentication headers in a GET request.

Example 7-17 REST service URL to list data sources

`https://<ioc_ihs>/ibm/ioc/api/datasource-service/datasources`

The response is shown in Example 7-18 (as obtained from a REST client for reference purposes). The response is a JavaScript Object Notation (JSON) object.

Example 7-18 Listing data sources sample response

```
1: {
  id: 19
  lastUpdateDate: 1383730718065
  messages: [0]
  type: "CSV"
  label:
    "{"group":"DataSourceI18n","key":"i18n_9c67f044-1052-46b9-814e-696aca0ca94b"}"
  i18nLabel: "Emergency events"
  description:
    "{"group":"DataSourceI18n","key":"i18n_5f8ee011-ffb5-422f-80a0-975d652aad2"}"
  i18nDescription: ""
  receiver: {...}-
  source: {...}-
  columns: [...10]-
  geometry: "POINT"
  routingLogic: "AND"
  styleLogic: "AND"
  archive: 1
  table: "IOC.TARGET_TABLE_19"
  view: "IOC.TARGET_VIEW_19"
  icon: "rescue.png"
  styles: [...1]-
  extensions: [...8]-
}-

2: {
  id: 22
  lastUpdateDate: 1383731236449
  messages: [0]
  type: "CSV"
  label:
    "{"group":"DataSourceI18n","key":"i18n_2cde4084-4cfc-41bd-849e-aaea6ff75f0f"}"
  i18nLabel: "EmergencyUnits"
  description:
    "{"group":"DataSourceI18n","key":"i18n_0953d31a-0476-43c4-85f1-a279626eebe8"}"
  i18nDescription: ""
  receiver: {...}-
  source: {...}-
  columns: [...8]-
  geometry: "POINT"
  routingLogic: "AND"
  styleLogic: "AND"
  archive: 1
  table: "IOC.TARGET_TABLE_22"
  view: "IOC.TARGET_VIEW_22"
  icon: "rescue.png"
  styles: [...1]-
  extensions: [...8]-
}
```

Parse the JSON response to retrieve the IDs for the data sources. In Example 7-18 on page 295, the IDs are 19 and 22. The IDs are used in the next steps.

Get the location for the first aid event

The collections resource of the spatial service is used to retrieve the location details for the first aid event with a given ID. Example 7-19 shows the service Uniform Resource Identifier (URI). Note that the service URI includes the data source ID (22) and the record Id (71) in the URL. The record ID is returned when the record was created.

Example 7-19 Get first aid event details URL

```
https://<ioc_>/ibm/ioc/api/spatial-service/collections/22/records/71
```

With the correct authorization headers, the response returned is shown in Example 7-20.

Example 7-20 First aid event details response

```
{
  id: 71
  annotationId: 0
  assesmentDateTime: null
  classification: null
  deleteFlag: false
  description: null
  endDateTime: {
    text: "Nov 22, 2013 11:30:00 AM"
    value: 1385137800000
  }-
  geometry: "POINT (-93.2591140 44.9730520)"
  lastUpdateDate: 1383732171000
  name: "Ambulance"
```

The *geometry* attribute contains the location properties for this first aid event.

Locate the emergency units near the first aid event

In this step, the spatial service feature resource is used to query the emergency units data source for units near the first aid event. The query is based on the geographical boundary parameters that can be constructed and adjusted to fit the business requirement.

Example 7-21 on page 297 shows the URI used for the query. The request parameters include a virtual boundary to search within. A new item can be added to the geospatial map by clicking More Actions → New Item on the Map page as described in “Adding an area or a line data item” at

http://pic.dhe.ibm.com/infocenter/cities/v1r6m0/topic/com.ibm.ioc.doc/use_mapaddarea.html.

The endpoints of the polygon drawn on the window by the operator can be sent as the boundaries to the spatial service call that is implemented by the IBM Intelligent Operations Center developer to handle custom correlation. See 7.5.3, “Finding nearby medical units” on page 290.

Example 7-21 Request URI to get units close to the first aid events

```
https://<ioc_ihs>/ibm/ioc/api/spatial-service/features/12?boundaries=POLYGON((-93.27319622040292%2044.9700734969083%2C-93.23925018310436%2044.9700734969083%2C-93.23925018310436%2044.975568679049346%2C-93.27319622040292%2044.975568679049346%2C-93.27319622040292%2044.9700734969083))
```

The response is a set of features. Each feature represents an emergency unit that is located inside the virtual boundary that is provided as shown in Example 7-22.

Example 7-22 Spatial service response

```
{
  type: "FeatureCollection"
  id: 22
  lastUpdateDate: 1383734462000
  features: [3]
  0: {
    type: "Feature"
    id: 71
    fid: 71
    deleted: false
    geometry: {
      type: "Point"
      coordinates: [2]
      0: -93.259114
      1: 44.973052
    }-
    properties: {
      DataSourceId: 22
      TenantId: null
      Classification: null
      AnnotationId: null
      Name: "Ambulance"
      TimeZoneOffset: -300
      EndDateTime: {
        text: "Nov 22, 2013 11:30:00 AM"
        value: 1385137800000
      }-
      DeleteFlag: 0
      LastUpdateDateTime: {
        text: "Nov 6, 2013 5:02:51 AM"
        value: 1383732171000
      }-
      ObjectId: 71
      Description: null
      AssessmentDateTime: null
      StartDateTime: {
        text: "Nov 22, 2013 10:30:00 AM"
        value: 1385134200000
      }-
    }-
  }-
}
```

The results returned by the spatial service can be displayed on the map to show the location of the emergency medical units, enclosed within the polygon.

Each item can be drilled down to view details. Each item can also be acted upon, like any data item, for example, to associate to a running standard operating procedure (SOP) or to run a custom action.



Developing mobile applications

This chapter describes how IBM Intelligent Operations Center capabilities can be extended to mobile devices.

This chapter shows through examples how to use IBM Intelligent Operations Center Representational State Transfer (REST) APIs. These APIs can help you to expose data and functionality to mobile applications built using IBM Worklight or other technologies for mobile applications.

This chapter includes the following topics:

- ▶ 8.1, “Technology overview” on page 300
- ▶ 8.2, “Use case overview” on page 302
- ▶ 8.3, “Implementation summary” on page 302
- ▶ 8.4, “Implementation details” on page 303
- ▶ 8.5, “Deploying the artifacts” on page 319
- ▶ 8.6, “Visualizing the results” on page 320

Note: For information about the sample code for the examples described in this chapter, see Appendix B, “Samples provided with this book” on page 383.

8.1 Technology overview

There are two IBM products needed for the implementation of the scenario in this chapter:

- ▶ IBM Intelligent Operations Center and its REST services
- ▶ IBM Worklight

The following sections provide an overview of the technologies used to implement the sample mobile application.

IBM Worklight Consumer Edition special note

IBM Intelligent Operations Center 1.6 includes the IBM Worklight Consumer Edition V6.0 software product. The IBM Intelligent Operations Center Premium User V1.6 license entitles you to the following functions:

- ▶ Customize mobile applications delivered with IBM Intelligent Operations Center
- ▶ Run mobile applications delivered with IBM Intelligent Operations Center provided that the Prohibited Components terms are met

Prohibited Components: Licensee is not authorized to use any of the following components or functions:

- ▶ Change project structure (of IBM Worklight Consumer Edition)
- ▶ Process data by the mobile application that does not originate directly from the Principal Program (of IBM Worklight Consumer Edition)
- ▶ Create a new project (of IBM Worklight Consumer Edition)

The IBM Intelligent Operations Center Premium User V1.6 license does not entitle you to use the included IBM Worklight Consumer Edition software for the following purposes:

- ▶ Develop new mobile applications
- ▶ Run mobile applications (other than those included with IBM Intelligent Operations Center)

You need to purchase an IBM Worklight license separately to be entitled to develop and run mobile applications.

8.1.1 IBM Intelligent Operations Center REST services

IBM Intelligent Operations Center REST services provide a set of uniform resource identifiers (URIs) that access data in IBM Intelligent Operations Center components, such as system properties and key performance indicators (KPIs). For more information about IBM Intelligent Operations Center REST services, see 1.3, “REST services” on page 6.

This chapter uses IBM Intelligent Operations Center REST services and in particular the following three services:

- ▶ Authentication service
- ▶ Spatial service
- ▶ Data injection service

Authentication service

IBM Intelligent Operations Center REST services can be used to validate the user's credentials. The service base URI is `/ibm/ioc/api/login-service/users`.

The service is called using Base64 encoding of the credentials with the authorization header.

To validate the result of the authentication process, the status code from the REST response is inspected. A status code of 200 means that the credentials are valid. The REST service returns an array of one JavaScript Object Notation (JSON) object that contains additional information about the user. A status code of 401 means that the credentials are not valid and the REST service returns an empty array.

Important note: The authentication service used in the sample code provided with this chapter is not supported and not documented at the time of writing this book. Check with your IBM representative for an update on the status of this service before you start developing mobile applications for IBM Intelligent Operations Center.

Spatial service

The spatial service provides access to the spatial-temporal records that are stored in IBM Intelligent Operations Center when a new data source is created. The items that are provided by this service are displayed on the map, on the location map, and in the list user interface (UI) components.

Using this service, you can query data records based on their geographical location. This service is used in the sample in this chapter to get a list of data source events that are happening within a certain geographical boundary. The service base URI is `/ibm/ioc/api/spatial-service`.

Data injection service

The data injection service is used to programmatically add, update, or cancel data items for a data source. The service base URI is `/ibm/ioc/api/data-injection-service`.

8.1.2 IBM Worklight

IBM Worklight is a mobile application development platform. It is designed to provide an open, comprehensive platform to build, run, and manage HTML5, hybrid, and native mobile apps for smartphones and tablets. It helps organizations of all sizes to efficiently develop, test, connect, run, and manage mobile applications using standards-based technologies and tools.

For more information, see the IBM Worklight product web page:

<http://www.ibm.com/software/products/us/en/worklight/>

Note: IBM Intelligent Operations Center 1.6 includes the IBM Worklight Consumer Edition V6.0 software product. For license restrictions, see “IBM Worklight Consumer Edition special note” on page 300.

IBM Worklight Developer Edition is available at no charge for evaluation purposes and can be downloaded at this website:

<http://www.ibm.com/developerworks/downloads/ws/worklight/>

8.2 Use case overview

During the festival in City A, the city official wants to efficiently address incidents that required first aid. The plan is to have operators onsite that can report the situations that require first aid as they occur.

The operators will be always on the move to cover more ground and they need to be able to file the first aid report and to request assistance using only their mobile devices. They also need to be able to use their mobile devices to see the situations happening in the surrounding areas using a map view.

8.2.1 Actors

The actors in this use case are the operators that are employed by the festival organizers to watch for situations that require first aid.

8.2.2 Desired outcome

There are three main goals that the solution described in this use case must achieve:

- ▶ Handle situations that require first aid efficiently during the event.
- ▶ Provide an easy reporting view of the situations in the area using a map component.
- ▶ Gather accurate statistics about how many situations that require first aid happened during the festival. This information will help city officials to be better prepared for other upcoming organized events.

8.2.3 IBM Intelligent Operations Center programming artifacts

Three IBM Intelligent Operations Center REST services are used in the implementation of this use case: spatial service, data injection service, and login service.

8.3 Implementation summary

Implementing the scenario in this chapter will follow these steps:

1. Create a new Worklight Project.
2. Build the IBM Worklight adapters that authenticate and connect to IBM Intelligent Operations Center REST services.
3. Build a mobile application that uses the adapters to get data from IBM Intelligent Operations Center. The data is used by the mobile application to display map locations. The mobile app also sends data to IBM Intelligent Operations Center to report a situation that requires first aid.
4. Test the mobile application with the mobile browser simulator.

8.3.1 Setting up the development environment

The development environment for this scenario requires IBM Worklight Studio, which is an Eclipse-based integrated development environment (IDE) that is used to develop mobile applications for IBM Worklight.

For information about system requirements, see *Detailed System Requirements for IBM Worklight and IBM Mobile Foundation*:

<http://www-01.ibm.com/support/docview.wss?uid=swg27024838>

For information about IBM Worklight installation and configuration, see *Installing and configuring*:

<http://bit.ly/1hsMZW2>

After the installation of IBM Worklight Studio is complete, mobile platform-specific tools are needed to build and deploy the mobile application to devices. For a complete listing of platforms and instructions, see the IBM Worklight V6.0 Information Center:

<http://bit.ly/1iR4952>

8.4 Implementation details

This section includes the step-by-step instructions for each implementation task.

8.4.1 Creating a new IBM Worklight project

An IBM Worklight project is a container that can include multiple mobile applications. It enables sharing resources and server components across mobile applications.

Create a new IBM Worklight project:

1. Click **File** → **New** → **Worklight Project**.
2. Type `ITSOMobile` as the name of the project, ensure that **Hybrid Applications** is selected, and click **Next** (see Figure 8-1 on page 304).

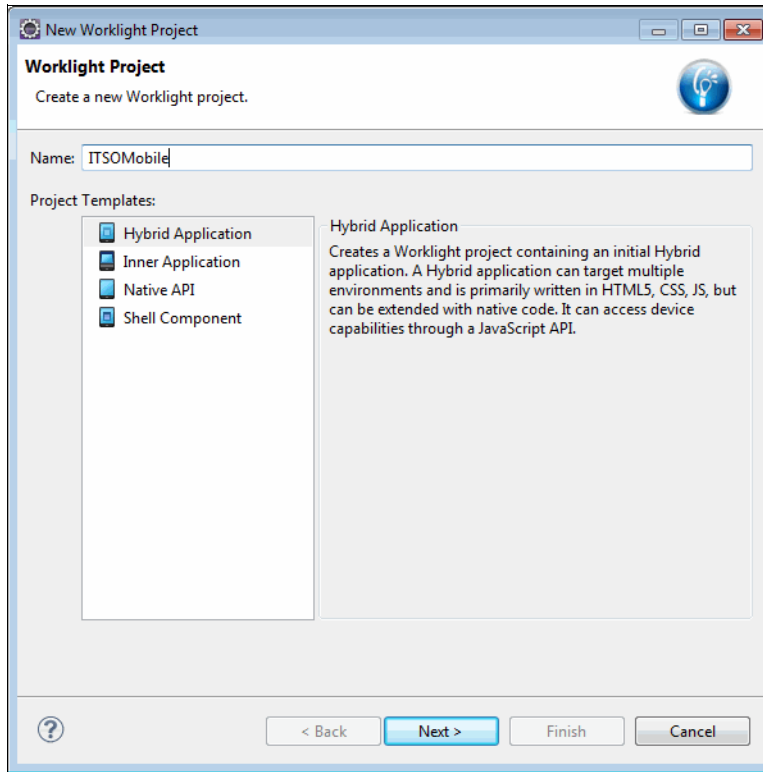


Figure 8-1 Creating a new IBM Worklight project

3. Type ITSOMobile in the Application Name field, select **Add Dojo Toolkit**, and click **Finish** (see Figure 8-2 on page 305).

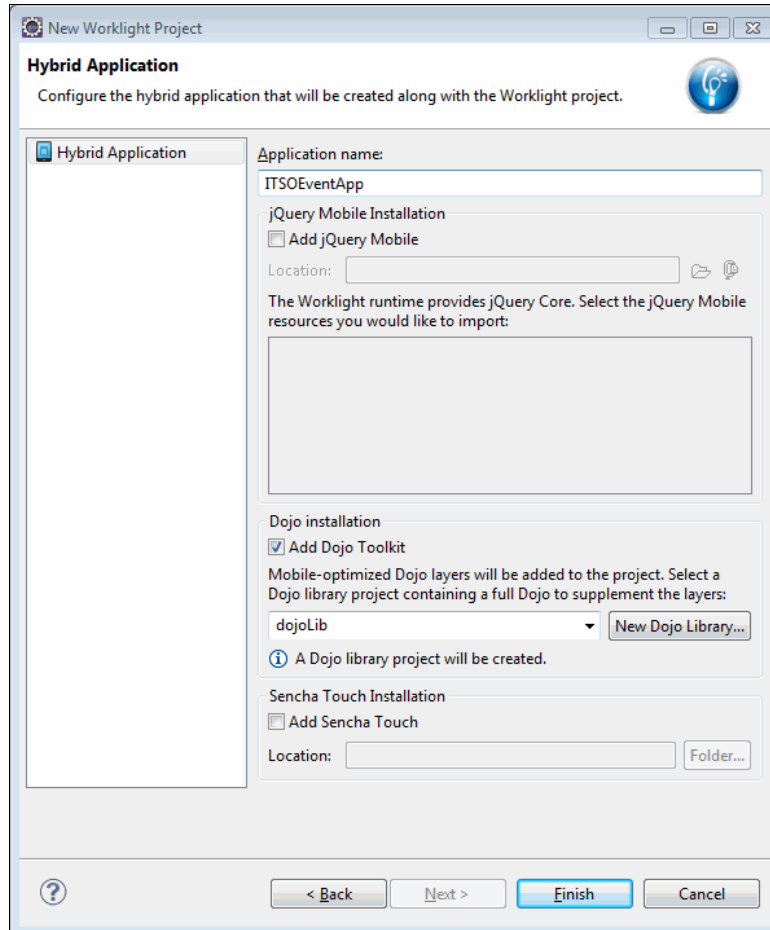


Figure 8-2 Create a new Worklight application

A new mobile application that is Dojo mobile-enabled is created.

Note: Dojo Mobile is a JavaScript framework that enables the rapid development of mobile web applications with a native interface. For more information about Dojo Mobile, see <http://dojotoolkit.org/features/mobile>.

8.4.2 Building the IBM Worklight adapters

IBM Worklight adapters are the server-side code for applications that are deployed on and serviced by IBM Worklight. Adapters connect to enterprise applications (in this use case, IBM Intelligent Operations Center) to deliver data to and from mobile applications and to perform any necessary server-side logic on this data.

Authentication adapter

The login REST service uses basic access authentication; therefore, the adapter must add the Authorization header to the HTTP request headers.

In this example, the authentication adapter has a single procedure called `authenticateUser()` that performs the following steps:

1. The mobile device calls the adapter procedure `authenticateUser`, passing to it the username and password.
2. The adapter procedure calls the login REST service setting the Authorization HTTP header with the username and password. The HTTP header is shown in Example 8-1. The string after the word `Basic` is the base64 encoding of the string `username:password`.

Example 8-1 HTTP header for authentication

```
Authorization: Basic QWxhZGRpbjpvYVUyIHNIc2FtZQ==
```

3. If the request is successful, the HTTP response status code is 200. The authentication Lightweight Third Party Authentication (LTPA) token is saved in the Worklight authentication realm.

The LTPA token in the authentication realm will be used in subsequent REST service calls.

Create a new adapter:

1. Click **File** → **New** → **Other** → **Worklight Adapter**.
2. The adapter that you create will connect to the IBM Intelligent Operations Center REST APIs over the HTTP protocol. Follow these steps:
 - a. For Adapter type, select **HTTP Adapter**.
 - b. In the Adapter name field, enter `Authentication Adapter`.
 - c. Click **Finish**.
3. Set up the Secure Sockets Layer (SSL) certificate keystore.

Before configuring the adapter XML file to connect to the IBM Intelligent Operations Center REST services, there is one preparation step needed. Because the adapter connects to the REST services securely using HTTPS, the IBM Intelligent Operations Center server security certificate must be imported to the Worklight server keystore.

Follow the steps described in the SSL certificate keystore setup at this website:

<http://bit.ly/Q7o0Qr>

Take note of the certificate *alias* that you define; it is used in the next step.

4. Configure the adapter to connect to the IBM Intelligent Operations Center REST services:
 - a. At the Adapter Editor, expand the **Connectivity** node and select **Connection Policy**.
 - b. In the Domain field, type the IBM Intelligent Operations Center server location.
 - c. Change the Protocol field to **https**. Set the Port field to the appropriate https port, which is typically 443.
 - d. From the Cookie policy list box, select **IGNORE_COOKIES**.
 - e. In the SSL Certificate Alias field, enter the alias that you defined in step 3.
5. Worklight creates two adapter procedures by default that must be removed. Delete the procedures named `getStories` and `getStoriesFiltered`.
6. Add a new procedure and name it `authenticateUser`.
7. Open the adapter implementation file `AuthenticationAdapter-impl.js`, remove the existing code, and add the code shown in Example 8-2 on page 307 to the file.

Example 8-2 Authenticate user procedure

```
var _Constants = {
    JSESSIONID : "JSESSIONID",
    SESSIONID : "com.ibm.ioc.sessionid",
    REALM : "IOCAuthRealm",
    LTPATOKEN : "LtpaToken2"
};

function authenticateUser(auth) {

    var input = {
        method : 'get',
        returnedContentType : 'application/json',
        path : "/ibm/ioc/api/login-service/users",
        headers : {
            Authorization : "Basic " + auth,
            Accept : "application/json"
        }
    };

    // get the JSESSIONID out of the request header
    var request = WL.Server.getClientRequest();
    var cookieHeader = request.getHeader("Cookie");
    var cookies = cookieHeader.split(";");
    var jSessionId = null;
    for ( var i = 0; i < cookies.length; i++) {
        var cookie = cookies[i];
        if (cookie.indexOf(_Constants.JSESSIONID) >= 0) {
            var valuePattern = /[=].*/;
            var value = "" + valuePattern.exec(cookie);
            jSessionId = value.trim().substr(1);
            break;
        }
    }

    if (jSessionId) {
        input.headers[_Constants.SESSIONID] = jSessionId;
        input.headers["Cookie"] = _Constants.JSESSIONID + "=" + jSessionId;
    }

    var result = WL.Server.invokeHttp(input);

    // if successful, set active user
    if (result.statusCode && result.statusCode == 200 && result.array) {
        var user = JSON.parse(JSON.stringify(result.array[0])); // clone
        user.base64 = auth;
        user.sessionId = null;
        user.ltpaToken = null;

        if (result.responseHeaders) {
            var responseHeaders = result.responseHeaders;

            // process the headers
            user.sessionId = responseHeaders[_Constants.SESSIONID];
        }
    }
}
```

```

// process the cookies
var setCookieHeader = responseHeaders["Set-Cookie"];
var cookieData = setCookieHeader.split(";");
for ( var i = 0; i < cookieData.length; i++) {
    var cookie = cookieData[i];
    if (cookie.indexOf(_Constants.LTPATOKEN) > -1) {
        var valuePattern = /[=].*/;
        var value = "" + valuePattern.exec(cookie);
        user.ltpaToken = value.trim().substr(1); // strip the
        // leading "="
        break;
    }
}

var userIdentity = {
    userId : user.uid,
    attributes : user
};

WL.Server.setActiveUser(_Constants.REALM, userIdentity);
return {
    statusMsg: 'authentication Success',
    statusCode : result.statusCode,
    userIdentity : userIdentity
};
} else {
WL.Server.setActiveUser(_Constants.REALM, null);
return {
    statusMsg: 'authentication failed, check logs for detailed error',
    statusCode : result.statusCode,
    userIdentity : {}
};
}
}
}

```

The code shown in Example 8-2 on page 307 gets the base64 encoded authorization from the mobile application and uses it to construct the authorization header. Then, the code sends an HTTP request to the login service and waits for the response to check whether the authentication is successful.

8. Configure authentication realms for the Worklight server. Expand the folder **server** → **conf** and open the file `authenticationConfig.xml`. Switch to the Source view to edit the XML content directly, then locate the `<realms>` element and add the realm in Example 8-3 to it.

Example 8-3 Adding the realm

```

<realm name="IOCAuthRealm" loginModule="NonValidatingLoginModule">
    <className>com.worklight.integration.auth.AdapterAuthenticator</className>
    <parameter name="login-function"
value="AuthenticationAdapter.authenticateUser" />
</realm>

```

- Then, locate the <loginModules> element and add the following login module to it (Example 8-4).

Example 8-4 Adding the login module

```
<loginModule name="NonValidatingLoginModule">
  <className>com.worklight.core.auth.ext.NonValidatingLoginModule</className>
</loginModule>
```

Events adapter

The events adapter is used to add a new emergency situation to the data source, or to retrieve a list of emergency situations in the area.

Create the events adapter:

- Repeat step 1 on page 303 through step 5 on page 306. Name the adapter FirstAidAdapter.
- Add two new procedures; name them getEvents and reportEvent. The Adapter Editor view is shown in Figure 8-3.

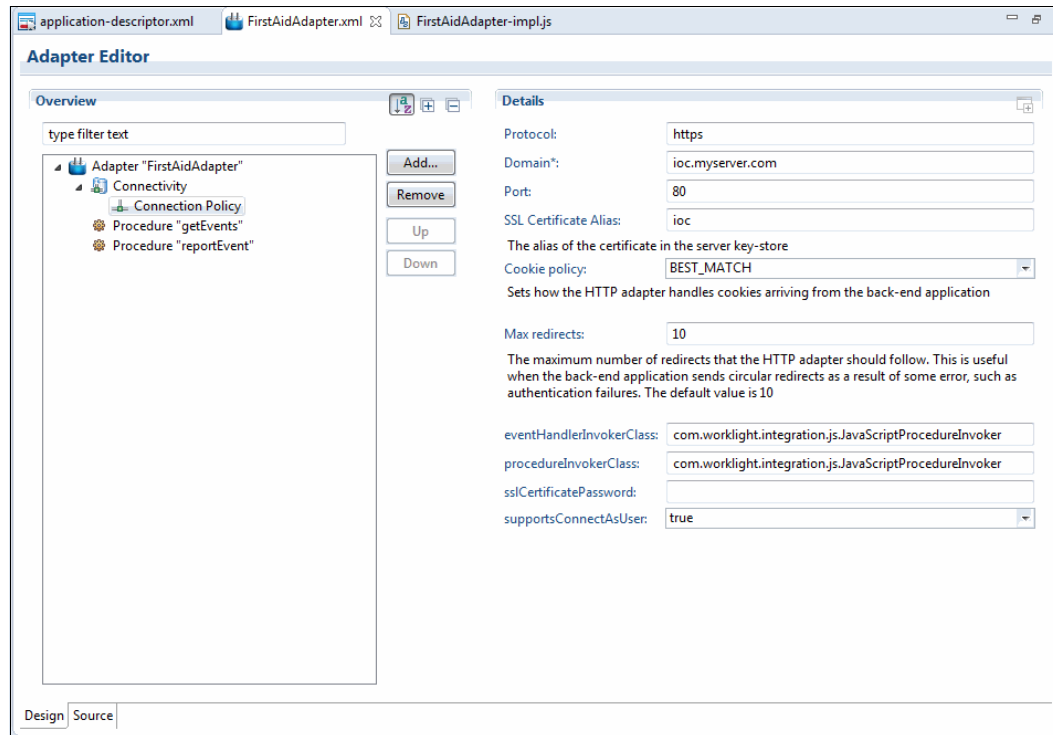


Figure 8-3 FirstAidAdapter - Adapter Editor view

3. Implement the adapter procedures:

Note: The sample code in this chapter uses the comma-separated value (CSV) data source `FirstAidEvents` created in 3.2.4, “CSV data source” on page 55. Make sure that you have created this data source.

Open the adapter script file **FirstAidAdapter-impl.js**. Remove the default implementation of the two functions `getStories` and `getStoriesFiltered`, and add code for the two procedures to be created:

- `getEvents` (see Example 8-5): This procedure gets a list of first aid events in the area. This procedure uses the IBM Intelligent Operations Center spatial service. It sends the data source Id, the geospatial boundaries to look within, and the LTPA token stored from the authentication adapter. This procedure returns a list of events (situations) that occurred within this geospatial boundary.

Example 8-5 getEvents procedure implementation

```
function getEvents(dataSourceId, boundaries, token) {
    path = "/ibm/ioc/api/spatial-service/features/" + dataSourceId;

    var user = WL.Server.getActiveUser(_Constants.REALM);
    httpHeaders = {
        Accept : "application/json",
        Cookie : _Constants.LTPATOKEN + "=" + user.attributes.ltpaToken
    };
    httpHeaders[_Constants.SESSIONID] = user.attributes.sessionId;
    httpHeaders["Cookie"] += "; " + _Constants.JSESSIONID + "="
        + user.attributes.sessionId;

    var input = {
        method : 'get',
        returnedContentType : 'json',
        path : path,
        headers : httpHeaders,
        parameters: {
            boundaries: boundaries,
            criterion: "deleteFlag=0"
        }
    };

    return WL.Server.invokeHttp(input);
}
```

- reportEvent (see Example 8-6). This procedure reports an event. This procedure uses the IBM Intelligent Operations Center data injection service to add the event data item. The procedure gets all minimal properties fields (start date, end date, location, and others) as a parameter. If adding the event is successful, it returns the created event object.

Example 8-6 reportEvent procedure implementation

```
function reportEvent(startDate, endDate, lastUpdate, location, name,
severity,
    sentBy, timezone, id, dataSourceId, token) {

    var body = {
        STARTDATETIME : startDate,
        ENDDATETIME : endDate,
        lastUpdateTime : lastUpdate,
        LOCATION : location,
        NAME : name,
        PROPERTY_2 : id,
        PROPERTY_3 : sentBy,
        PROPERTY_4 : severity,
        TIMEZONEOFFSET : timezone
    };
    var strBody = JSON.stringify(body);

    path = "/ibm/ioc/api/data-injection-service/datablocks/" + dataSourceId
        + "/dataitems";

    var user = WL.Server.getActiveUser(_Constants.REALM);
    httpHeaders = {
        Accept : "application/json",
        Cookie : _Constants.LTPATOKEN + "=" + user.attributes.ltpaToken
    };
    httpHeaders[_Constants.SESSIONID] = user.attributes.sessionId;
    httpHeaders["Cookie"] += "; " + _Constants.JSESSIONID + "="
        + user.attributes.sessionId;

    var input = {
        method : 'post',
        returnedContentType : 'json',
        path : path,
        headers : httpHeaders,
        body : {
            content : strBody,
            contentType : "application/json"
        }
    };

    return WL.Server.invokeHttp(input);
}
```

Implementing the application UI

This section shows an example of a simple application to address the use case requirements. The mobile app is developed using Dojo mobile widgets.

Creating the Log In view

The application starts with a login window that takes the username and password and invokes the AuthenticationAdapter to authenticate the user with IBM Intelligent Operations Center.

If the authentication is successful, the LTPA token is automatically retrieved from the adapter and stored in the IBM Worklight authentication realm. The design of the login window is shown in Figure 8-4.

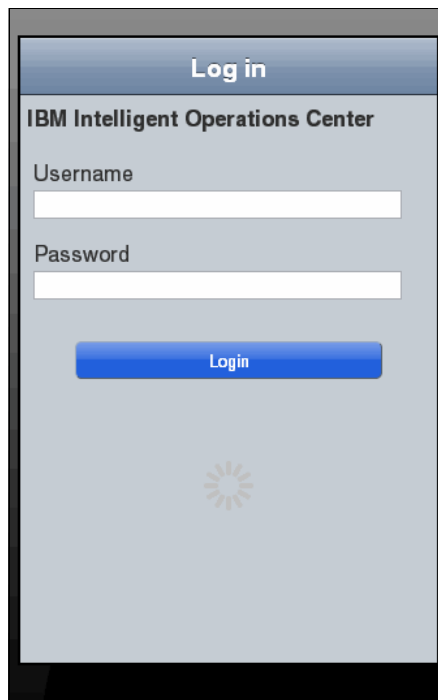


Figure 8-4 Login window

When the user enters the credentials and clicks **Login**, the following steps occur:

1. The username and password are collected from the form and encoded using base64 encoding as shown in Example 8-7.

Example 8-7 Encoding username and password

```
var username = this.dapUsername.value;
var password = this.dapPassword.value;

var auth = window.btoa(username + ":" +
password);
```

2. The authentication adapter is invoked using the encoded authentication as shown in example Example 8-8.

Example 8-8 Invoking the authentication adapter

```
var invocationData = {
    adapter : 'AuthenticationAdapter',
    procedure : 'authenticateUser',
    parameters : [ auth ]
};

WL.Client.invokeProcedure(invocationData, {
    onSuccess : lang.hitch(this, this.onAuthSuccess),
    onFailure : lang.hitch(this, this.onError),
});
```

3. If the authentication is successful, the onAuthSuccess function is called with the adapter response, and it transitions to the application home view as shown in Example 8-9.

Example 8-9 Successful authentication

```
if (result.status == 200) {
    if (result.invocationResult.isSuccessful) {
        this.performTransition('IndexView', -1, 'slide');
    } else {
        // not authenticated
        this.dap_LoginStatus.innerHTML = "Login failed. Invalid username
or password.";
    }
}
```

Creating the home view

The Home page (home view) contains a simple menu with three options: Report Event, View First Aid Events, and Settings. The design of the page is shown in Figure 8-5 on page 314.

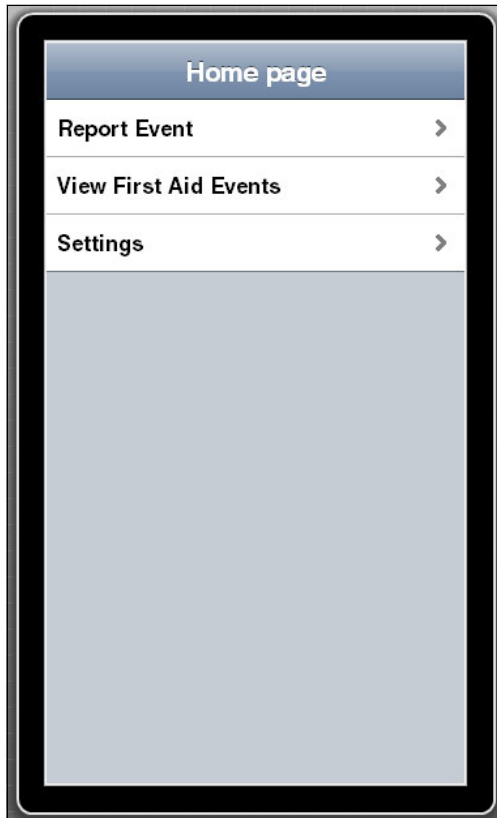


Figure 8-5 Mobile app Home page

Creating the Settings view

The Settings view is a simple view where a user can add settings for two items:

- ▶ Event sender name: This setting is used in the `sentBy` property when a new event is created.
- ▶ The second setting is the data source ID. This sample application uses the Emergency Events data source created in 3.2.4, “CSV data source” on page 55 with the sample file `EmergencyUnits.csv`.

The ID of the data source must be retrieved and set in this setting before using the application.

Note: The data source ID can be obtained by querying the data source service to get a list of data sources and their IDs. The REST service URL is shown:

```
https://<ioc_ibs>/ibm/ioc/api/datasource-service/datasources
```

For the complete details, see “Identify data sources” on page 294.

When Save is clicked, the two settings are saved to the `localStorage`, as shown in Example 8-10.

Example 8-10 Saving settings

```
save: function(){
    localStorage.senderName = this.dapSenderName.value;
    localStorage.dataSourceId = this.dapDataSourceId.value;
    this.performTransition('IndexView', -1, 'slide');
}
```

Creating the Report Event view

The UI for reporting an event is straightforward. The user types the emergency title, the emergency description, and the emergency severity. Then, the user clicks **Report Event**. The view is shown in Figure 8-6.



Figure 8-6 Report Event view

When Report Event is clicked, the `FirstAidAdapter` event procedure `reportEvent` is called by the code shown in Example 8-11.

Example 8-11 Adding a new event from the mobile application

```
// prepare parameters
var name = this.dapTitle.value;
var description = this.dapDescription.value;
var severity = this.dapSeverity.value;
// start date is current date
var startDate = stamp.toISOString(new Date());
// end date is current date + one week
var endDate = stamp.toISOString(date.add(new Date(), "week", 1));
var lastUpdate = "";
// Location is fixed for now, but it can easily be detected from the
```

```

// mobile device location
var location = "POINT(-93.25676500799601 44.97392549261744)";
// Sender name is set from settings view
var sender = localStorage.senderName || "";
var id = "2";
var timeZone = "0";

var dataSourceId = localStorage.dataSourceId || "19";

var invocationData = {
  adapter : 'FirstAidAdapter',
  procedure : 'reportEvent',
  parameters : [ startDate, endDate, lastUpdate, location, name,
    severity, sender, timeZone, id, dataSourceId]
};

WL.Client.invokeProcedure(invocationData, {
  onSuccess : lang.hitch(this, this.onReportEventSuccess),
  onFailure : lang.hitch(this, this.onReportEventError)
});

```

Note: This example uses a hardcoded location for the new event data. From a mobile device, it is easy to acquire and use the current location instead. For more information, see *geolocation.getCurrentPosition* at this website:

<http://bit.ly/1iTXEhE>

Creating the Events view

The Events view uses the events adapter procedure to get a list of events that occur within the given geospatial boundary. It then displays the data in either a list format or on a map. This approach is similar to the approach that is used in the IBM Intelligent Operations Center Operations view.

When the view is initialized, it calls the *getEvent* procedure from the code that is shown in Example 8-12.

Example 8-12 Calling the *getEvent* procedure

```

var boundaries = "POLYGON((-93.27319622040292 44.9700734969083,-93.23925018310436
44.9700734969083,-93.23925018310436 44.975568679049346,-93.27319622040292
44.975568679049346,-93.27319622040292 44.9700734969083))";
var dataSourceId = localStorage.dataSourceId;
var invocationData = {
  adapter : 'FirstAidAdapter',
  procedure : 'getEvents',
  parameters : [ dataSourceId, boundaries]
};

WL.Client.invokeProcedure(invocationData, {
  onSuccess : lang.hitch(this, function(result){
// console.log(F, "invokeProcedure::success:: " +
response.invocationResult.item);
    this._buildEventsList(result.invocationResult.features);
    this.dap_progressInd.stop();
  }),

```

```
        onFailure : lang.hitch(this, function(){
            console.error(F, "invokeProcedure::failure:: " +
response.invocationResult);
            this.dap_progressInd.stop();
        }),
    });
```

When the events are retrieved, they are initially displayed in a list format as shown in Figure 8-7. Clicking any of the events opens a detailed view.

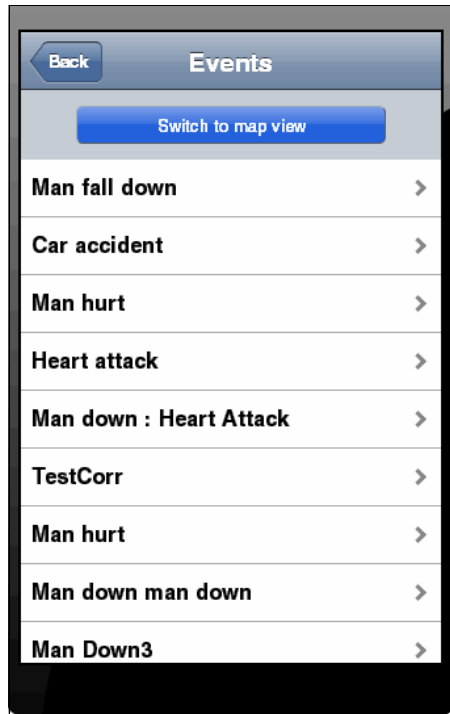


Figure 8-7 Events list

The user can also view this list of events using a map. Clicking **Switch to map view** displays the events on a map as shown in Figure 8-8 on page 318.

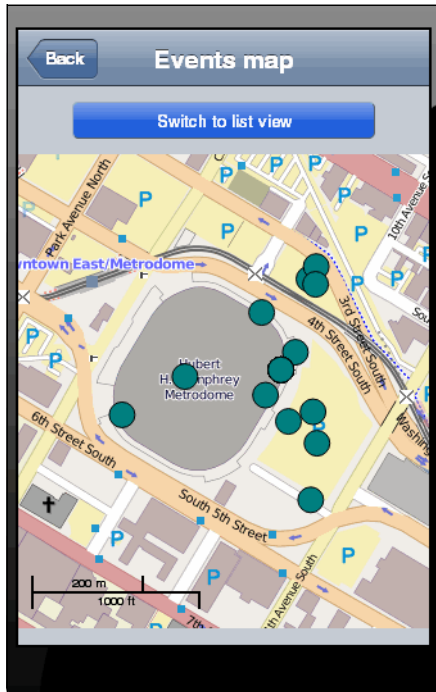


Figure 8-8 Map view

The map is built using the Dojo-wrapped widget of the OpenLayers maps.

Note: Dojo's `dojox.geo.openlayers` module provides a mapping component based on the OpenLayers library. This component is mainly a wrapper of the OpenLayers library with additional features, such as displaying a geo-referenced GFX shape and placing geo-referenced widgets on a background map. For more information, see the `dojox.geo.openlayers` documentation:

<https://dojotoolkit.org/reference-guide/1.9/dojox/geo/openlayers.html>

Example 8-13 shows the code for initializing the map and creating markers for events.

Example 8-13 Initializing the map and creating event markers

```

initMap : function() {
    this.openMap = new Map("map");
    var layer = new GfxLayer();
    for(var i = 0; i < this.eventsData.length; i++){
        this.createMarker(this.eventsData[i], layer);
    }
    this.openMap.addLayer(layer);
    // map center point
    var pt = {
        latitude : 44.9733145,
        longitude : -93.256639
    };
    // Fitting map to the center point and setting the extent around it
    this.openMap.fitTo({
        position : [ pt.longitude, pt.latitude ],
        extent : 0.011
    });
},

```



```
createMarker : function(event, layer) {
    var p = new Point({
        x : event.geometry.coordinates[0],
        y : event.geometry.coordinates[1]
    });
    var f = new GeometryFeature(p);
    f.setFill([ 0, 128, 128 ]);
    f.setStroke([ 0, 0, 0 ]);
    f.setShapeProperties({
        r : 10
    });
    layer.addFeature(f);
},
```

8.5 Deploying the artifacts

This section describes how to deploy the programming artifacts described in this use case.

8.5.1 Building and deploying mobile devices

When you develop mobile applications, you must install and use specialized tools that depend on the operating system for which you develop the applications, such as iOS or Android. For more information, see *Installing mobile specific tools*:

<http://bit.ly/1uwEToc>

8.5.2 Deploying server-side components

When you build a Worklight project that contains applications in IBM Worklight Studio, a WAR file and a set of .wlapp files are created in the `bin` folder of your IBM Worklight project. The next step is to deploy the project and the application to test and production environments:

- ▶ Deploy server-side components to the local test environment

To deploy the IBM Worklight application to the local test server, right-click the **IBM Worklight application**, select **Run as** → **Build all and deploy**.

The Worklight environment is built.

- ▶ Deploy server-side components to the production environment

To deploy the IBM Worklight Application on a remote production server, take the Worklight artifacts created in the `bin` folder of your IBM Worklight project and deploy them on the application server. For detailed information, see *Deploying IBM Worklight applications to test and production environments*:

<http://bit.ly/QbBXgm>

8.6 Visualizing the results

IBM Worklight includes a mobile browser simulator that can be used to test the application. Access the mobile browser simulator:

1. Right-click the **ITSOEventApp** application in the Project Explorer view and select **New** → **Worklight Environment**. Select the platform environments that you want to add and click **Finish**.
2. Build and deploy the application to the mobile browser simulator.
Right-click the Worklight app (**ITSOEventApp**) and select **Run as** → **Build all and deploy**. Wait for the build and deployment process to complete.
3. Open the mobile browser simulator. Right-click the Worklight project and select **Open Worklight Console**.
4. In the Worklight Console, click the environment name that was created in step 1 to open the mobile browser simulator. The application opens and the login window opens as shown in Figure 8-9.

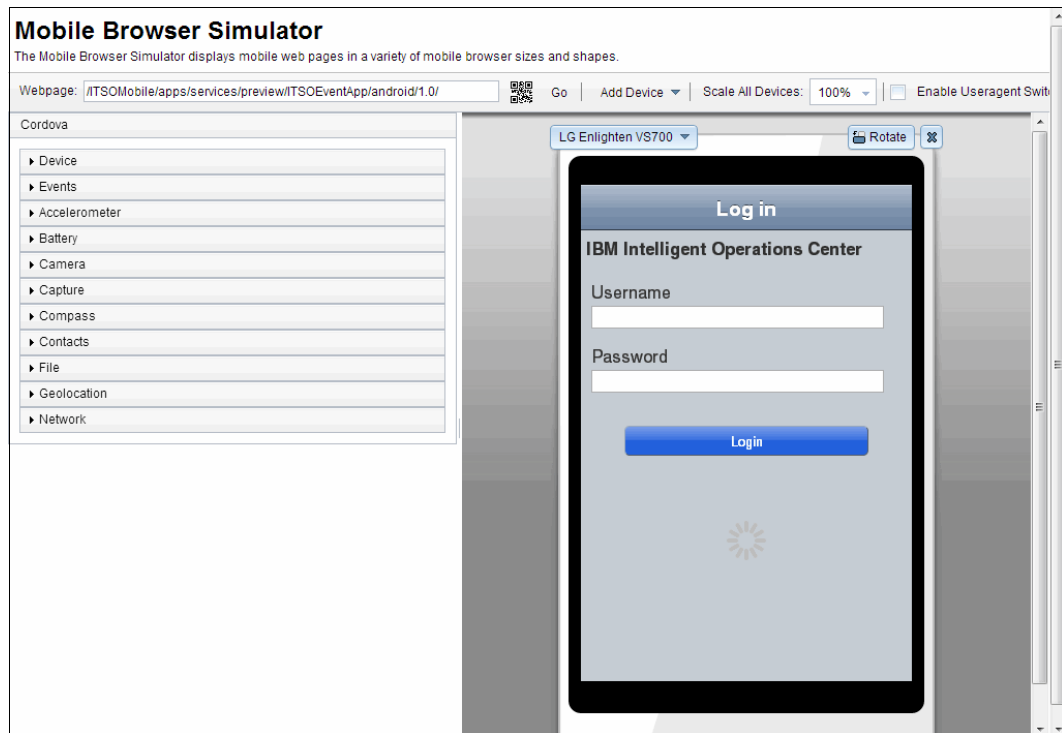


Figure 8-9 The sample application in the mobile browser simulator



Integrating with external systems

This chapter describes the integration of IBM Intelligent Operations Center with other systems to share data and to visualize and act on events. This chapter lists the options available for integration with external systems and provides implementation examples.

This chapter includes the following topics:

- ▶ 9.1, “Overview” on page 322
- ▶ 9.2, “Integration with IBM SmartCloud Control Desk through SOP automation activity” on page 322
- ▶ 9.3, “Integration through REST service activity” on page 328
- ▶ 9.4, “Routing data to the integration topic” on page 331
- ▶ 9.5, “Using an external service URL” on page 335
- ▶ 9.6, “Other options for integration” on page 348

Note: For information about the sample code for the examples described in this chapter, see Appendix B, “Samples provided with this book” on page 383.

9.1 Overview

The following examples illustrate the need to integrate the IBM Intelligent Operations Center solution with external systems:

- ▶ Road work data from the public utilities databases needs to be pulled in to identify the roads that are currently undergoing maintenance and that might not be available to citizens during the city events.
- ▶ A work order must be created in an external system (for example, IBM SmartCloud Control Desk) to put first response units on standby mode for handling a public safety emergency.

There are several ways in which integration with external systems can be implemented:

- ▶ Integration through standard operating procedures (SOPs):
 - Integrating with IBM SmartCloud Control Desk through the Automation Activity type in an SOP
 - Integrating with external systems through the Representational State Transfer (REST) Activity type in an SOP
- ▶ Using a data source routing expression to direct events to a predefined integration topic. External applications listening on the topic can pick the events up for processing.
- ▶ Using a service URL on the data source REST service to pull in data from external systems for visualization on the map or for list views.
- ▶ Using server-side Java services.
- ▶ Creating a custom data source action from the preview card.

9.2 Integration with IBM SmartCloud Control Desk through SOP automation activity

During the festival at City A, the crowd size is continuously monitored. An SOP is triggered when the crowd size exceeds a predefined threshold. The SOP takes action to mitigate the crowd size and manage any possible emergencies. To implement this use case, the SOP connects to the city's asset management system, IBM SmartCloud Control Desk, which is external to IBM Intelligent Operations Center. It triggers the job plan to prepare personnel on standby mode in case an emergency occurs that requires an increase in the public safety personnel.

9.2.1 Actors

The actors of this use case are operators of the agencies and organizations involved in defining, approving, and implementing SOPs. For more information, see 5.4.1, "Actors" on page 155.

9.2.2 Desired outcome

The desired outcome is to enable multiple agencies to collaborate while responding to incidents. Agencies need to be able to integrate the IBM Intelligent Operations Center SOP capabilities with their own systems to complete actions in response to events and incidents.

The job plan in IBM SmartCloud Control Desk to put personnel on standby mode is launched from IBM Intelligent Operations Center.

9.2.3 IBM Intelligent Operations Center programming artifacts

The artifact is the SOP that launches the IBM SmartCloud Control Desk job plan.

9.2.4 Implementation summary

The following steps summarize the implementation of this use case:

1. Configure IBM Intelligent Operations Center for connection to IBM SmartCloud Control Desk.
2. Verify IBM SmartCloud Control Desk settings for web service access.
3. Create the SOP that invokes the creation of an IBM SmartCloud Control Desk job plan using an Automation Activity type.

9.2.5 Implementation details

The following sections describe the steps to implement this use case.

Configure the connection to IBM SmartCloud Control Desk

Configure IBM Intelligent Operations Center for connection with IBM SmartCloud Control Desk:

1. Configure the system properties in IBM Intelligent Operations Center for connection to IBM SmartCloud Control Desk.

Click **System Administration** → **System Properties** (see Figure 9-1 on page 324).

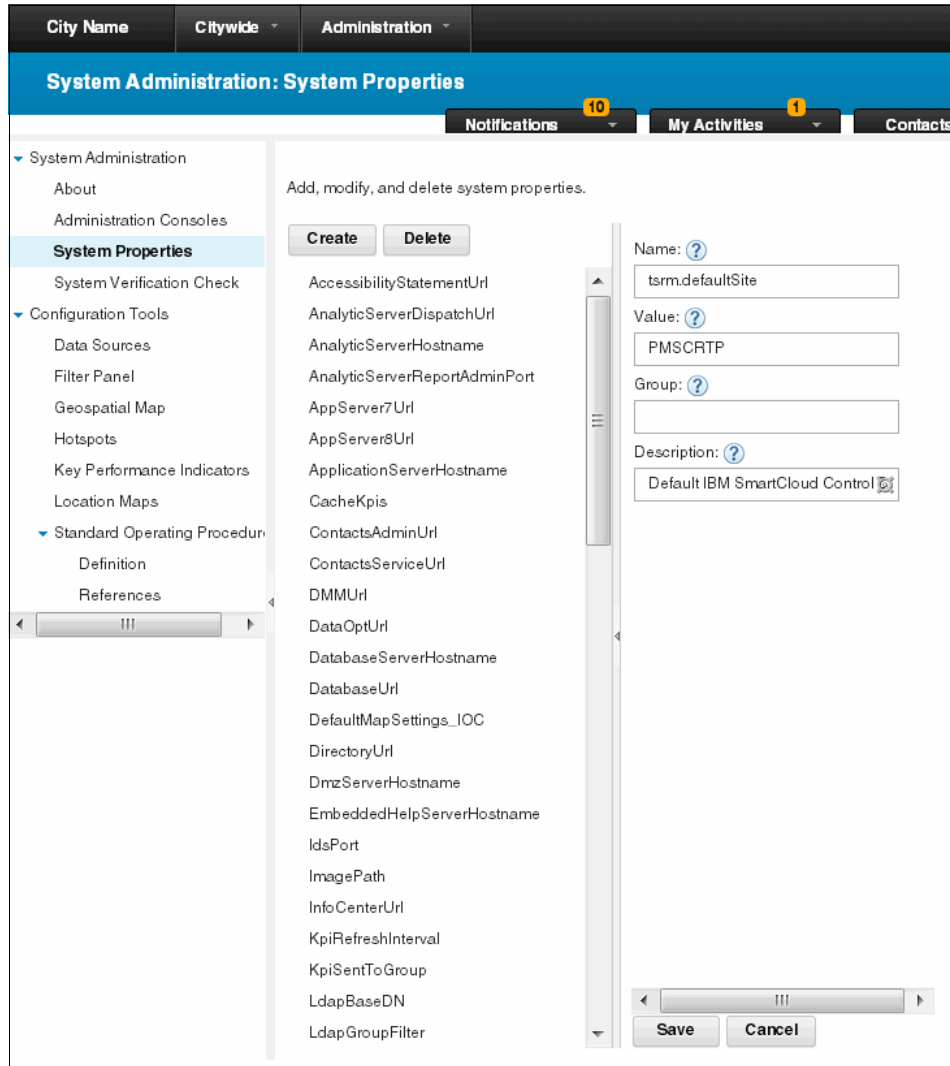


Figure 9-1 Updating the System Properties for IBM SmartCloud Control Desk

2. Update the values for the properties shown in Table 9-1. If a property is not present, create it.

Table 9-1 IBM SmartCloud Control Desk-specific properties in System Properties

Property name	Property default value	Description
tsrm.defaultSite	PMSCRTP	IBM SmartCloud Control Desk default site
tsrm.endPointProcEngine	<tsrm_installation_URI>/meaweb/	IBM SmartCloud Control Desk integration endpoint
tsrm.listAutomation.whereClause	JPNUM LIKE 'PLUSI%'	Query for clause to get automations
tsrm.startAutomation.workflowName	APPFWFO	IBM SmartCloud Control Desk workflow name

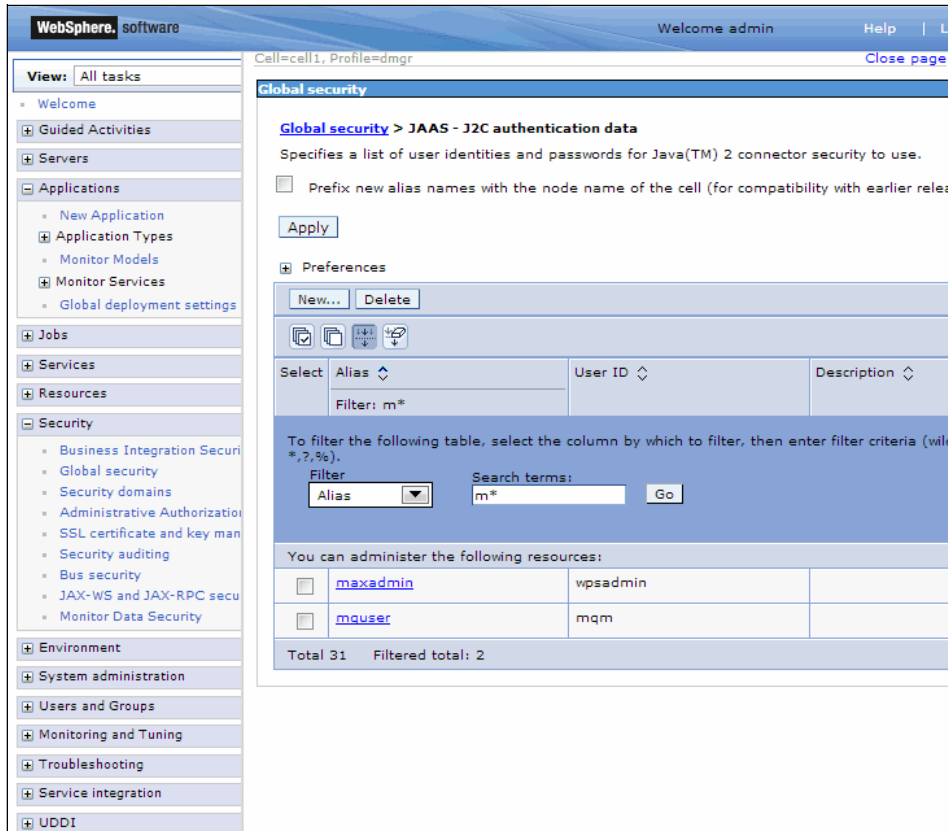


Figure 9-2 J2C authentication alias for the IBM SmartCloud Control Desk administrator

9.2.6 Verify IBM SmartCloud Control Desk settings for web service access

Verify that the available job plan list in IBM SmartCloud Control Desk can be accessed through a web service:

1. Verify that the web service is successfully deployed in IBM SmartCloud Control Desk.

Using a browser that is running locally on the IBM SmartCloud Control Desk server (integration endpoint for IBM SmartCloud Control Desk), enter the following URL:

```
http://<tsrm_server>:82/meaweb/os/PLUSIMXJOBPLAN
```

The following message is returned:

```
Servlet is running. Please use HTTP POST to post data.
```

2. Verify that the SOP REST service can access IBM SmartCloud Control Desk.

In your browser, enter the following URL:

```
http://<server:port>/ioc_er_sopba_rest/activity/0/listautomations
```

A message similar to the one shown in Figure 9-3 on page 327 is displayed.


```

{"PLUSIRESPO":"Evacuate impacted areas (Respond and Recover)&#x200E;","PLUSIPREPA":"Prepare for severe weather evacuation (Prepare)&#x200E;","PLUSIPREPS":"Prepare personnel for standby (Prepare)","PLUSIMITIG":"Initial preparation for severe weather (Mitigate)&#x200E;"}

```

Figure 9-3 Message returned by IBM SmartCloud Control Desk web service

9.2.7 Create the SOP to trigger the job plan through automation activity

Follow these steps:

1. In the IBM Intelligent Operations Center Solution Administration view, click **Configuration Tools** → **Standard Operating Procedures** → **Definition**.
2. In the Create Activity Definition window, select **Automation Activity** from the Activity type list (see Figure 9-4).

Figure 9-4 Selecting the Automation Activity type in the SOP activity definition

3. Select the job plan of interest from the Job plan drop-down list (Figure 9-5 on page 328).

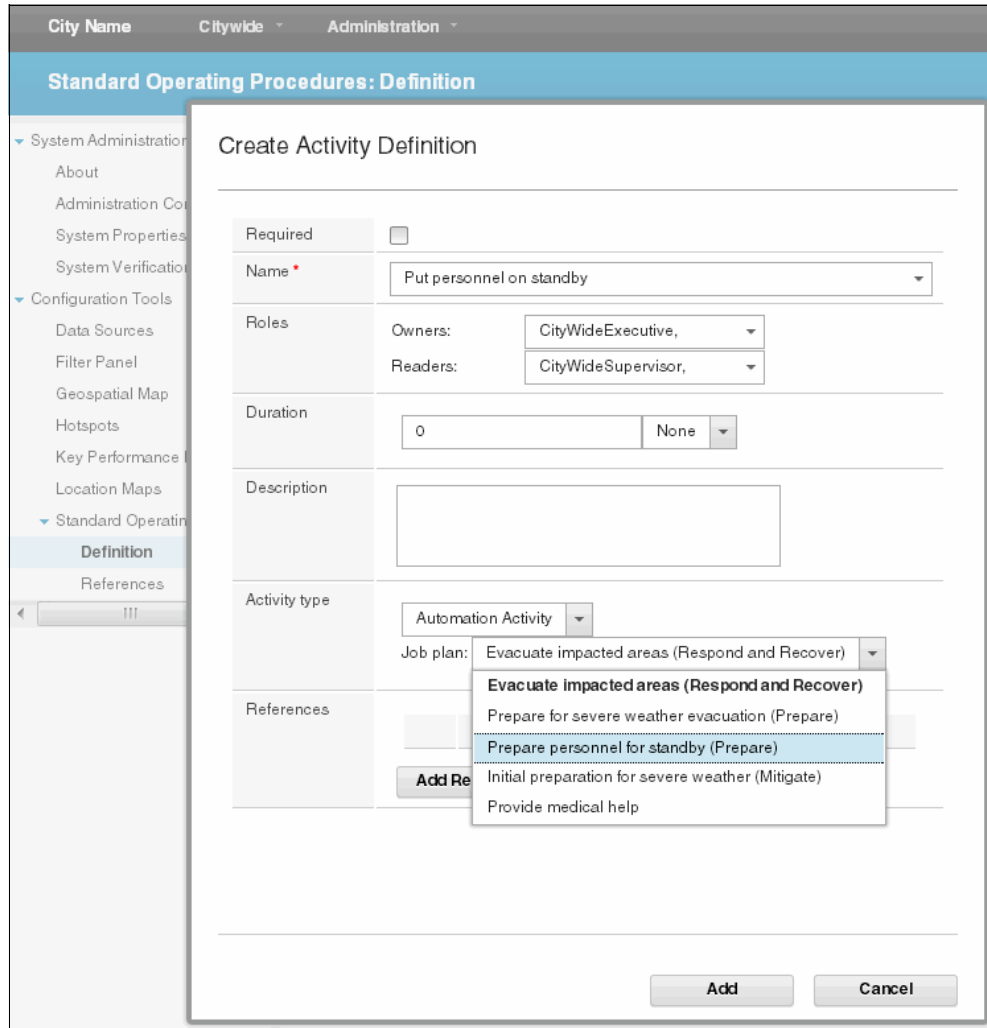


Figure 9-5 Selecting the job plan

For more information, see 5.2.4, “Configuring automation activities” on page 132 and 5.6, “Manage crowd incident SOP” on page 177.

9.3 Integration through REST service activity

You can use the REST Service Activity type to create a REST service call and specify a REST service URL:

1. In the Create Activity Definition window, select **REST Service** from the Activity type pull-down list (see Figure 9-6 on page 329).

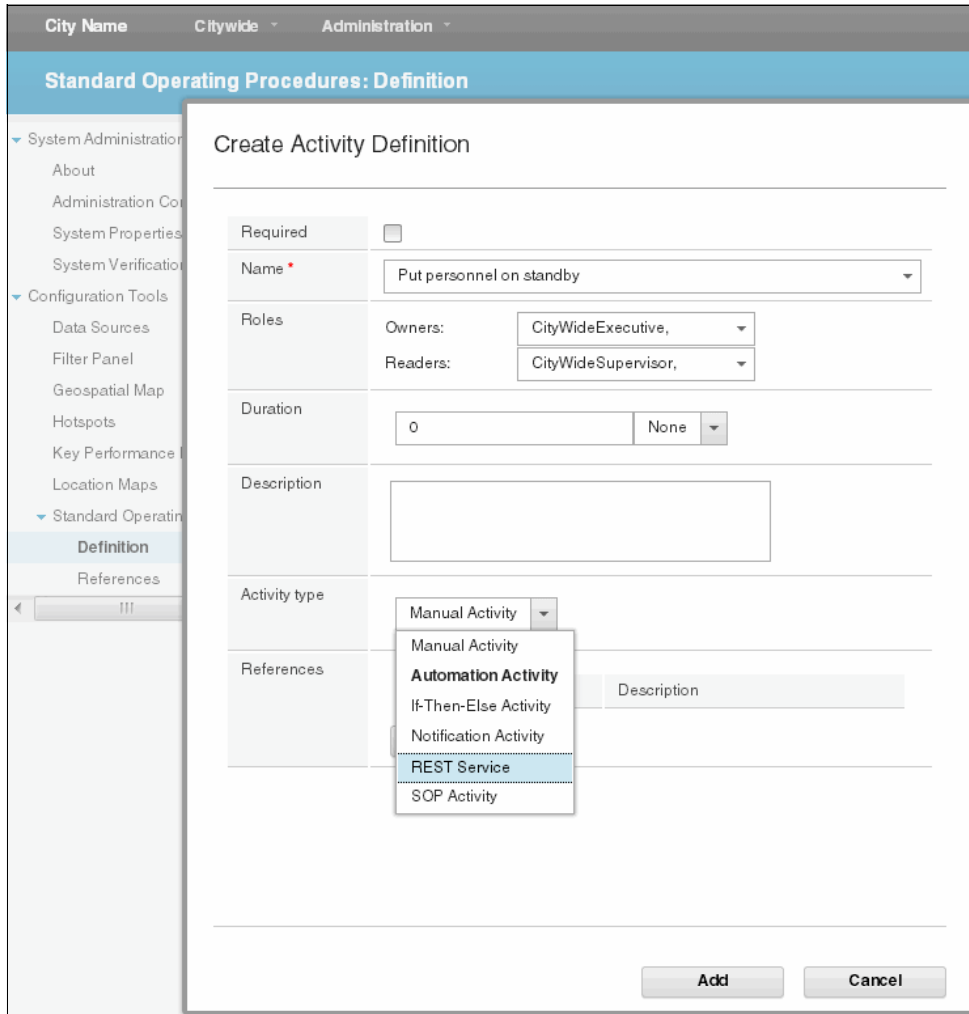


Figure 9-6 Selecting REST Service activity type

2. Enter the REST service URL and credentials to access the service in the remote system as shown in Figure 9-7 on page 330.

The screenshot shows a 'Create Activity Definition' dialog box. The 'Required' checkbox is checked. The 'Name' field contains 'Dispatch security teams to support locally'. Under 'Roles', 'Owners' is set to 'CityWideOperator' and 'Readers' is set to 'CityWideAdmin'. The 'Duration' is set to '0' and 'None'. The 'Description' field is empty. Under 'Activity type', 'REST Service' is selected. The 'REST Service' field contains 'http://<host>:<port>/sampl'. The 'User ID' is 'waswebadmin' and the 'Password' is masked with asterisks. At the bottom, there is an 'Add Reference' button and a table with columns 'Name' and 'Description'. At the very bottom of the dialog are 'Add' and 'Cancel' buttons.

Figure 9-7 Specify REST Service URL details

A complete SOP with both Automation and REST Service activities is shown in Figure 9-8 on page 331.

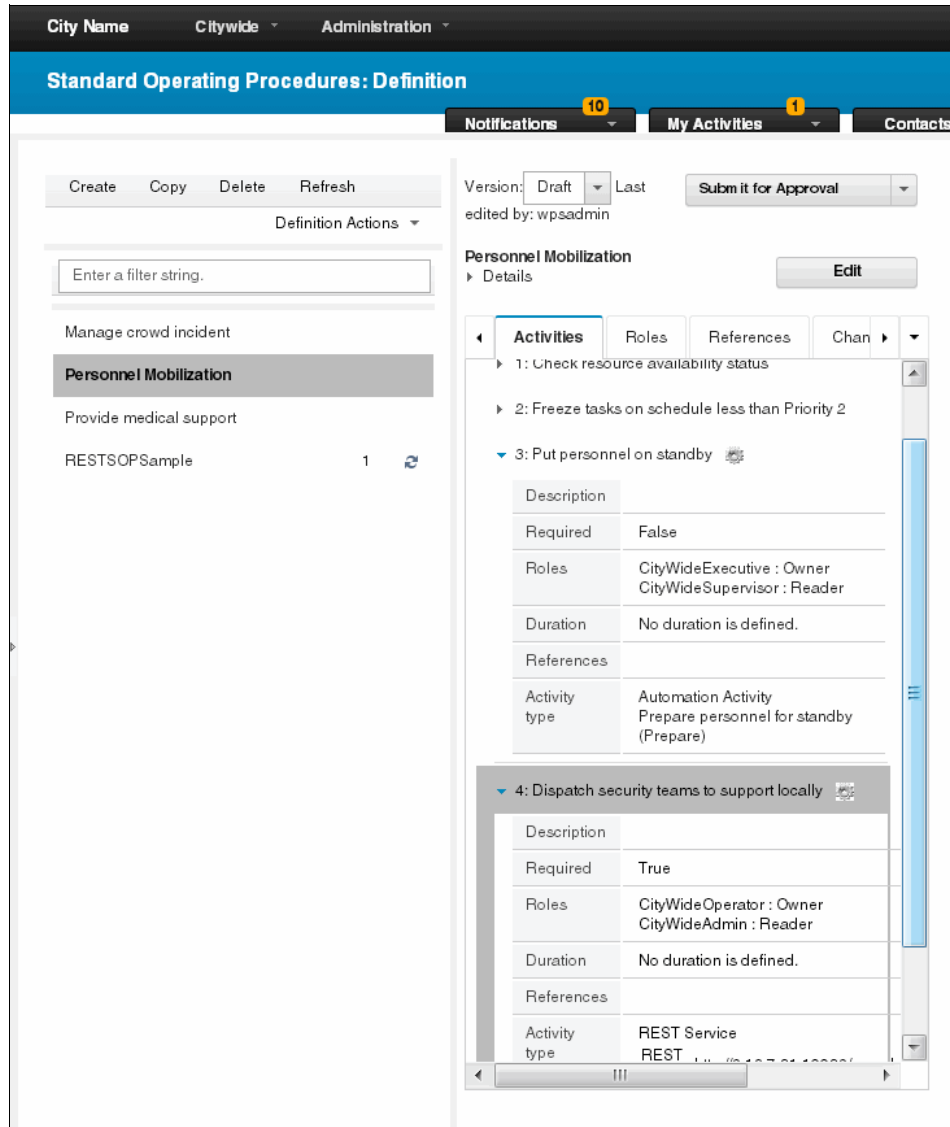


Figure 9-8 SOP showing Automation and REST Service activities

For more information, see 5.2.7, “Configuring REST service activities” on page 141.

9.4 Routing data to the integration topic

During the festival at City A, a couple of medical emergency events occur.

A celebrity is spotted, and a crowd rapidly gathers. In the chaos, a man falls down, cuts his foot, and bleeds profusely. Bystanders notify the festival operator. The festival operator creates a new event for the incident on the map. The festival operator uses IBM Intelligent Operations Center to locate the nearest emergency unit and directs them to the location of the casualty. For a complete use case description, see 7.3, “Use case overview” on page 285.

The information entered by the operator indicates the severity of the emergency event as *high*. This results in the emergency event information automatically being routed to the medical unit system, helping them to arrive prepared to treat the injured person.

In this use case, the integration to a system external to IBM Intelligent Operations Center takes place through a routing mechanism that directs the data item fulfilling the filter criteria to a predefined integration topic.

9.4.1 Desired outcome

This use case has the following goals:

- ▶ Dispatch the appropriate first aid emergency units quickly to provide the necessary assistance so that the festival continues without disruption.
- ▶ When they arrive at the scene, the medical units are equipped and ready to handle the medical emergency.

9.4.2 IBM Intelligent Operations Center programming artifacts

IBM Intelligent Operations Center can determine whether an incoming data item must be routed to an integration topic and then send the data for processing to a predefined WebSphere Application Server integration topic.

The following programming artifacts are used to implement this use case:

- ▶ Message-driven bean (MDB), activation specification, and the integration topic
A Java Message Service (JMS) activation specification is associated with one or more MDBs and provides the configuration necessary for them to receive messages. The integration topic is the topic on which the MDB is listening. IBM Intelligent Operations Center predefines the integration topic `jms/iocIntegration`.
- ▶ Data sources
- ▶ Routing schema

9.4.3 Implementation summary

This section uses the sample projects `sample_ioc_integration_mdb` and `sample_ioc_integration_EAR`. The `sample_ioc_integration_mdb` project contains the MDB that listens on the integration topic.

The following steps summarize the implementation of this use case:

1. Create an activation specification for the integration topic.
2. Create and deploy an MDB to listen on the integration topic and process incoming XML messages, as needed.
3. Configure a data source with a routing expression to direct data items to the integration topic.

9.4.4 Implementation details

The following sections describe the steps to implement this use case.

Create an activation specification for the integration topic

Follow these steps from the WebSphere Application Server 8.0 Deployment Manager console for WebSphere Portal:

1. Click **Resources** → **JMS** → **Activation Specification**.
2. Click **New** to add an activation specification.
Accept the default scope of **cell=cell1** and the **default message provider**, and click **OK**.
3. On the Configuration tab, complete the general properties form for the activation specification as shown in Table 9-3.

Table 9-3 Creation of activation specification for the integration topic

Name	Java Naming and Directory Interface (JNDI) name	Destination	Destination JNDI name	Bus name
topicTestAS	eis/topicTestAS	Topic	jms/iocIntegration	IOC.cell1.BUS

4. Click **Apply** → **OK**.
5. Save to the master configuration, and restart the clusters.

Create and deploy an MDB

Follow these steps:

1. Create an MDB to listen on the integration topic `jms/iocIntegration`.
See *Developing an enterprise application to use message-driven beans* at this website: <http://bit.ly/1jA0Hsz>
Optionally, import the MDB project sample into your IBM Rational Application Developer workspace using the Import Existing Project option.
2. Export the application ear.
3. Deploy the MDB (application ear created) to the Application Server:
 - a. Click **Solution Administration** → **System Administration** → **Administration Consoles**.
 - b. Click **Application Server**.
 - c. Log in with your administration user ID and password.
 - d. Add the new application to the application server by stepping through the New Enterprise Application wizard:
 - i. Click **Applications** → **New Application** → **New Enterprise Application**.
 - ii. Select the path to the new application on a local or remote file system and browse to locate and upload the file.
 - iii. Choose the fast path option to install.
 - iv. At the **Map modules to servers** step, select the **PortalCluster** from the list, and click **Apply**.
 - v. Click **Next** until you reach the summary step and then click **Finish**.

Add a routing expression to direct data items to the integration topic

In a data source, the routing expression can be configured to filter and direct data items to the predefined integration topic. Other applications can listen to the predefined integration topic to capture the data item and process it, as necessary.

Follow these steps:

1. On the Routing tab for the data source, click **Add Routing Expression**.
2. Choose the property to compare, the operation, and enter the criteria. In this use case, data items with severity equal to High are selected as shown in Figure 9-9.

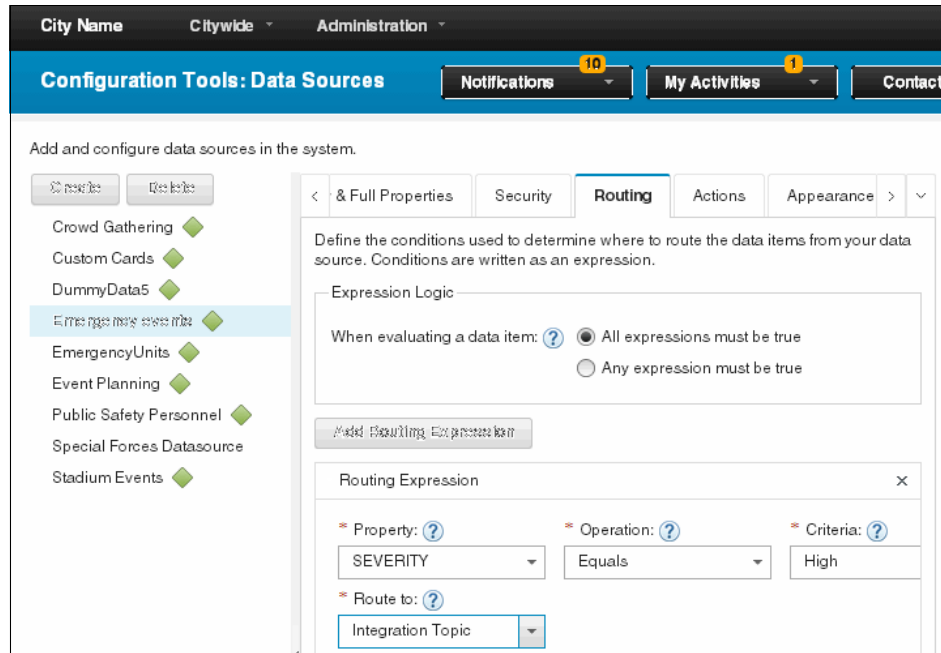


Figure 9-9 Routing data items to the predefined integration topic

3. Click **Save**.

When a data item satisfies the criteria specified in the routing expression, it is published to the predefined integration topic.

The routing schema file includes the minimum properties. Update it to include the key and full properties that are specific for the data source that you want to route to the integration topic, as shown in Figure 4-8 on page 77 and Example 4-1 on page 77. The routing schema file can then be used to generate a set of Java Architecture for XML Binding (JAXB) classes. Applications that are listening to the integration topic can use the routing schema file to convert an incoming XML message to Java classes that the applications can then process.

Note: Interim fix PO02174 provides updates to the IBM Intelligent Operations Center database and to the IBM Intelligent Operations Center V1.6.0.1 routing schema. For more information, see the IBM Intelligent Operations Center database and routing schema updates at this website:

<http://www-01.ibm.com/support/docview.wss?uid=swg24036674>

When the operator enters the emergency event data item into IBM Intelligent Operations Center (as described in 7.5.2, “Creating emergency events” on page 287), the routing expression evaluates the Severity property value of the emergency event entered. Because the operator has marked the event as high severity, the data item is published to the integration topic on which the medical unit system is listening. The first responders will have the information necessary to arrive prepared to handle the emergency.

Figure 9-10 shows the XML message published to the integration topic.

```

- <iocrt:record xsi:schemaLocation="urn:ibm:industry:solutions:ioc:routing:1.6 IOC-routing-v1.6.xsd ">
  <iocrt:DATASOURCEID>19</iocrt:DATASOURCEID>
  <iocrt:SEVERITY>High</iocrt:SEVERITY>
  <iocrt:MESSAGE>
    Man fell down and cut his foot, bleeding, needs immediate medical attention.
  </iocrt:MESSAGE>
  <iocrt:ID>214</iocrt:ID>
  <iocrt:SOURCE>Spotter</iocrt:SOURCE>
  <iocrt:NAME>TEST Integration Topic Publish</iocrt:NAME>
  <iocrt:TIMEZONEOFFSET>-300</iocrt:TIMEZONEOFFSET>
  <iocrt:LOCATION>POINT (-93.2583744 44.9750829)</iocrt:LOCATION>
  <iocrt:ENDDATETIME>2013-11-21T21:15:00+00:00</iocrt:ENDDATETIME>
  <iocrt:DELETEFLAG>>false</iocrt:DELETEFLAG>
  <iocrt:LASTUPDATEDATETIME>2013-11-15T04:33:11+00:00</iocrt:LASTUPDATEDATETIME>
  <iocrt:OBJECTID>229</iocrt:OBJECTID>
  <iocrt:STARTDATETIME>2013-11-08T21:00:00+00:00</iocrt:STARTDATETIME>
  <iocrt:ASSESSMENTDATETIME>2013-11-08T21:00:00+00:00</iocrt:ASSESSMENTDATETIME>
</iocrt:record>

```

Figure 9-10 Message published to predefined integration topic

9.5 Using an external service URL

When the data in an external system cannot be brought into the IBM Intelligent Operations Center because it is not in the standard form, or due to other reasons, a service can be created to pull in data from that data source in the GeoJSON format. The URL for such a service is referred to as an *external service URL*. It is specified in the service property in the request content on the datasource-service POST request. The data source that is created is referred to as the *external service data source*.

The external service URL is a property of the data source configuration. If specified on a data source configuration, and the IBM Intelligent Operations Center client code calls the service to fetch the data items for the data source. The returned data items are displayed on the IBM Intelligent Operations Center visualization layer, including maps, location maps, a list, and charts. Routing expressions cannot be applied to external service data sources.

The external service URL is implemented as a REST service.

The custom data source must be defined via the data source API and the service property specifies the custom service URL as shown in Example 9-1 on page 336.

Example 9-1 Service URL example

```
"geometry": "POINT",
"archive": 1,
"service":
"https://<ioc_ihs>/sample/custom/service/url/sample-service/features",
"icon": "311.png",
"extensions": [{
  "name": "category",
  "type": "CUSTOM",
  "disabled": false,
  "multiple": false,
  "module": "ICPO",
```

In this use case, knowing the location of special forces is important for event management, in case they are required to aid in emergencies. The information is not in standard form that can be brought into IBM Intelligent Operations Center as a data source. Therefore, the service that provides information about the location of special forces is included as an external service URL.

9.5.1 Custom service requirements

The custom service must satisfy the following conditions:

- ▶ The custom service must be accessible.
- ▶ The custom service must return data in the same format as the spatial service features resource GeoJSON¹ feature collection format, as shown in Example 9-2.

Example 9-2 Feature collection format

```
{
  "type": "FeatureCollection",
  "id": 2,
  "lastUpdateDate": 1384987567928,
  "features": [{
    "fid": 1,
    "geometry": {
      "type": "Point",
      "coordinates": [-93.33701133728287,
        45.02243088817261]
    },
    "properties": {
      "DatasourceId": 2,
      "EndDateTime": {
        "value": 1384464641963,
        "text": "2013-11-14 16:30:41.963441"
      },
      "Property_1": "Special Forces Unit 3",
      "LastUpdateDateTime": {
        "value": 1384464641963,
        "text": "2013-11-14 16:30:41.963441"
      },
      "ObjectId": 1,
      "Name": "Special Forces Unit 3",
```

¹ For more information about GeoJSON, see *Feature Collection Objects* at this website: <http://geojson.org/geojson-spec.html#feature-collection-objects>

```

    "TimeZoneOffset": -360,
    "StartDateTime": {
      "value": 1384464641963,
      "text": "2013-11-14 16:30:41.963441"
    }
  },
  "deleted": false,
  "type": "Feature",
  "id": 1
},
{
  "fid": 2,
  "geometry": {
    "type": "Point",
    "coordinates": [-93.23195457458759,
44.97946204601823]
  },
  "properties": {
    "DatasourceId": 2,
    "EndDateTime": {
      "value": 1384464641963,
      "text": "2013-11-14 16:30:41.963441"
    },
    "Property_1": "Special Forces Unit 12",
    "LastUpdateDateTime": {
      "value": 1384464641963,
      "text": "2013-11-14 16:30:41.963441"
    },
    "ObjectId": 2,
    "Name": "Special Forces Unit 12",
    "TimeZoneOffset": -360,
    "StartDateTime": {
      "value": 1384464641963,
      "text": "2013-11-14 16:30:41.963441"
    }
  },
  "deleted": false,
  "type": "Feature",
  "id": 2
},
{
  "fid": 3,
  "geometry": {
    "type": "Point",
    "coordinates": [-93.30559730529649,
44.95031229770789]
  },
  "properties": {
    "DatasourceId": 2,
    "EndDateTime": {
      "value": 1384464641963,
      "text": "2013-11-14 16:30:41.963441"
    },
    "Property_1": "Special Forces Unit 2",
    "LastUpdateDateTime": {

```

```

        "value": 1384464641963,
        "text": "2013-11-14 16:30:41.963441"
    },
    "ObjectId": 3,
    "Name": "Special Forces Unit 2",
    "TimeZoneOffset": -360,
    "StartDateTime": {
        "value": 1384464641963,
        "text": "2013-11-14 16:30:41.963441"
    }
},
"deleted": false,
"type": "Feature",
"id": 3
}],
{
    "fid": 4,
    "geometry": {
        "type": "Point",
        "coordinates": [-93.21135520934635,
            44.95249904262329]
    },
    "properties": {
        "DatasourceId": 2,
        "EndDateTime": {
            "value": 1384464641963,
            "text": "2013-11-14 16:30:41.963441"
        },
        "Property_1": "Special Forces Unit 31",
        "LastUpdateDateTime": {
            "value": 1384464641963,
            "text": "2013-11-14 16:30:41.963441"
        },
        "ObjectId": 4,
        "Name": "Special Forces Unit 31",
        "TimeZoneOffset": -360,
        "StartDateTime": {
            "value": 1384464641963,
            "text": "2013-11-14 16:30:41.963441"
        }
    },
    "deleted": false,
    "type": "Feature",
    "id": 4
}
]]
}

```

Feature collection properties

The feature collection returned has the properties listed in Table 9-4.

Table 9-4 Properties of the feature collection

Name	Type	Description
id	Integer	Data source ID
type	String	Collection type
features	Array	Array of data items returned by query
lastUpdateDate	Long	Timestamp indicating when the collection was last updated
messages	String array	List of messages optionally returned by the service

Feature properties

Feature collection consists of features with the properties defined as shown in Table 9-5.

Table 9-5 Properties of the feature item

Name	Type	Description	
id	String	String representation of the ObjectID minimum property	
fid	String	String representation of the ObjectID minimum property	
type	String	Data item type	
deleted	Boolean	Indicates whether the data item has been canceled	
geometry	Object	Geospatial location information in GeoJSON format	
	type	String	Geospatial type (for example, Point or Line String)
	coordinates	Array	Collection of longitude/latitude values
properties	Object	<p>An object that contains the minimum and key properties. The object always includes the following properties:</p> <ul style="list-style-type: none"> DATASOURCEDID TenantId Classification AnnotationId Name TimeZoneOffset EndDateTime DeleteFlag ObjectId LastUpdatedDateTime AssessmentDateTime StartDateTime Description 	

Header, path, and query parameters

The parameters for the HTTP header are shown in Table 9-6 on page 340.

Table 9-6 HTTP header parameters

Name	Type	Description
Accept-Language	String	Specifies the acceptance response media types (application/json)
User-TimeZone	Integer	Timezone offset in minutes

The parameters for the path are shown in Table .

Table 9-7 Path parameters

Name	Type	Description
datasourceld	Integer	The id of the data source to run the query against

Table 9-8

The query parameters, which are shown in Table 9-9, must be handled by the service URL.

Table 9-9 Query parameters

Name	Type	Description		
locale	String	The language locale		
boundaries	String	The geospatial boundary of the query specified in well-known text (WKT) format		
buffer	Double	The number of kilometers that the geospatial boundary (boundaries parameter) is expanded before executing the query		
criterion	String	Additionally filter criteria using the following keys:		
		Key	Operator	Value
		startDateTime	=, >, <, >=, <=	ISO 8601 formatted date time value
		endDateTime	=, >, <, >=, <=	ISO 8601 formatted date time value
		lastUpdatedDateTime	=, >, <, >=, <=	ISO 8601 formatted date time value
		deleteFlag	=	0 or 1, indicates whether to include canceled data items in the query results
		Minimum, Key, or Full property name		
onlyOn	String	JavaScript Object Notation (JSON) string representing a range or a holiday as defined using the Filter Panel configuration tool and returned by the filter-service holidays resource.		

9.5.2 Adding an external service URL

This section uses the sample projects `sample_ioc_rest_service_url_setup`, `sample_ioc_rest_service_url_web`, and `sample_ioc_rest_service_url_web_ear`. The `sample_ioc_rest_service_url_web` project contains the Java service that provides the external service URL. The `sample_ioc_rest_service_url_setup` project contains the associated database script and records.

Add an external service URL:

1. Install the web module (WAR file) or the enterprise application archive (EAR file) containing the web module in your application server.
2. Verify that you can access the REST service. The example uses this REST service:
`https://<ioc_ihs>/sample/custom/service/url/sample-service/features/`
3. Optional: Define the translation strings for the data source name and data source column names (see 6.6.4, "Using the translation widget and internationalization service" on page 248).
4. Define the data source using the data source service API with a POST request to the `http://<ioc_ihs>/ibm/ioc/api/datasource-service/datasources` URI. Example 9-3 shows the JSON payload for the POST call.

Example 9-3 Sample payload for creating a data source from a service URL

```
{
  "type": "Database",
  "label":
  "{\\"group\\":\\"DataSourceI18n\\",\\"key\\":\\"i18n_service_url_example_datasource_name\\"}",
  "source": {
    "protocol": "com.ibm.ioc.datasource.ColumnParserDatabase"
  },
  "columns": [{
    "type": "TIMESTAMP",
    "sourceType": "TIMESTAMP",
    "sourceName": "STARTDATETIME",
    "targetNames": ["START_DATE_TIME"],
    "hint": "STARTDATETIME",
    "label":
    "{\\"group\\":\\"DataSourceI18n\\",\\"key\\":\\"i18n_service_url_example_col_1\\"}",
    "category": "MINIMAL",
    "format": "TIMESTAMP",
    "length": "26",
    "scale": "6",
    "updatable": false
  },
  {
    "type": "TIMESTAMP",
    "sourceType": "TIMESTAMP",
    "sourceName": "ENDDATETIME",
    "targetNames": ["END_DATE_TIME"],
    "hint": "ENDDATETIME",
    "label":
    "{\\"group\\":\\"DataSourceI18n\\",\\"key\\":\\"i18n_service_url_example_col_2\\"}",
    "category": "MINIMAL",
    "format": "TIMESTAMP",
```

```

        "length": "26",
        "scale": "6",
        "updatable": false
    },
    {
        "type": "VARCHAR",
        "sourceType": "VARCHAR",
        "sourceName": "LOCATION",
        "targetNames": ["LOCATION"],
        "hint": "LOCATION",
        "label":
"{\"group\": \"DataSourceI18n\", \"key\": \"i18n_service_url_example_col_3\"}",
        "category": "MINIMAL",
        "format": "SHAPE",
        "length": "30",
        "updatable": true
    },
    {
        "type": "VARCHAR",
        "sourceType": "VARCHAR",
        "sourceName": "NAME",
        "targetNames": ["NAME"],
        "hint": "NAME",
        "label":
"{\"group\": \"DataSourceI18n\", \"key\": \"i18n_service_url_example_col_4\"}",
        "category": "MINIMAL",
        "format": "STRING",
        "length": "30",
        "updatable": true
    },
    {
        "type": "TIMESTAMP",
        "sourceType": "TIMESTAMP",
        "sourceName": "LASTUPDATEDATETIME",
        "targetNames": ["LAST_UPDATE_DATE_TIME"],
        "hint": "LASTUPDATEDATETIME",
        "label":
"{\"group\": \"DataSourceI18n\", \"key\": \"i18n_service_url_example_col_5\"}",
        "category": "MINIMAL",
        "format": "TIMESTAMP",
        "length": "26",
        "scale": "6",
        "updatable": false
    },
    {
        "type": "INTEGER",
        "sourceType": "INTEGER",
        "sourceName": "TIMEZONEOFFSET",
        "targetNames": ["TIMEZONE_OFFSET"],
        "hint": "TIMEZONEOFFSET",
        "label":
"{\"group\": \"DataSourceI18n\", \"key\": \"i18n_service_url_example_col_6\"}",
        "category": "MINIMAL",
        "format": "INTEGER",
        "length": "10",

```



```

    "updatable": false
  },
  {
    "type": "VARCHAR",
    "sourceType": "VARCHAR",
    "sourceName": "DESCRIPTION",
    "targetNames": ["Property_1"],
    "hint": "DESCRIPTION",
    "label":
    "{\\"group\\":\\"DataSourceI18n\\",\\"key\\":\\"i18n_service_url_example_col_7\\"}",
    "category": "KEY",
    "length": "100",
    "updatable": true
  },
  {
    "type": "VARCHAR",
    "sourceType": "VARCHAR",
    "sourceName": "ADDRESS",
    "targetNames": ["Property_2"],
    "hint": "ADDRESS",
    "label":
    "{\\"group\\":\\"DataSourceI18n\\",\\"key\\":\\"i18n_service_url_example_col_8\\"}",
    "category": "FULL",
    "length": "100",
    "updatable": true
  }
]},
"geometry": "POINT",
"archive": 1,
"service":
"https://<ioc_ihs>/sample/custom/service/url/sample-service/features",
"icon": "311.png",
"extensions": [{
  "name": "category",
  "type": "CUSTOM",
  "disabled": false,
  "multiple": false,
  "module": "ICPQ",
  "associated": "1"
}],
{
  "name": "name",
  "type": "CUSTOM",
  "disabled": false,
  "multiple": false,
  "module": "ICPQ",
  "associated": "Service URL Example"
}],
{
  "name": "Cancel",
  "type": "ACTION",
  "disabled": false,
  "multiple": false,
  "pathLocation": "/ibm/ioc/theme/static/js/js",
  "pathName": "ioc",
  "module": "ioc/com/ibm/ioc/moreactions/Cancel"
}

```

```

    },
    {
      "name": "NewItem",
      "type": "ACTION",
      "disabled": false,
      "multiple": false,
      "module": "IOC"
    },
    {
      "name": "Update",
      "type": "ACTION",
      "disabled": false,
      "multiple": false,
      "pathLocation": "/ibm/ioc/theme/static/js/js",
      "pathName": "ioc",
      "module": "ioc/com/ibm/ioc/moreactions/Edit"
    }
  ]
}

```

Refer to the *IBM Intelligent Operations Center v1.6 REST APIs* document, chapter “Data source service” and section *Post method* for more details.

5. Optionally, add authorizations for the data source using the authorization service as shown in Example 9-4.

Example 9-4 Adding authorization for the data source

```

https://<ioc_ihs>/ibm/ioc/api/authorization-service/authorizations/
{
  "resource_id": 277,
  "resource_type": "datasource",
  "identity": "CityWideOperator",
  "identity_type": "G",
  "write_access": true
}

```

6. Add the data source to a filter pane:
 - a. Navigate to **Administration** → **System Administration** → **Configuration Tools** → **Filter Panel**.
 - b. Choose one of the existing filter panes or create a new one.
 - c. Move the newly created data source from the list of available data sources to the data sources included on this pane to make it available on the operator view.

Note: For more information about adding a data source to a filter pane, see the video tutorials:

- ▶ Creating a DB2 data source at http://youtu.be/khnCVLq_wCE
- ▶ Creating a CSV data source at <http://youtu.be/4fnr7w-hDE>
- ▶ Creating a Common Alerting Protocol (CAP) data source at <http://youtu.be/00mQ8phuSsq>

A new filter pane for the service URL data source just created is shown in Figure 9-11.

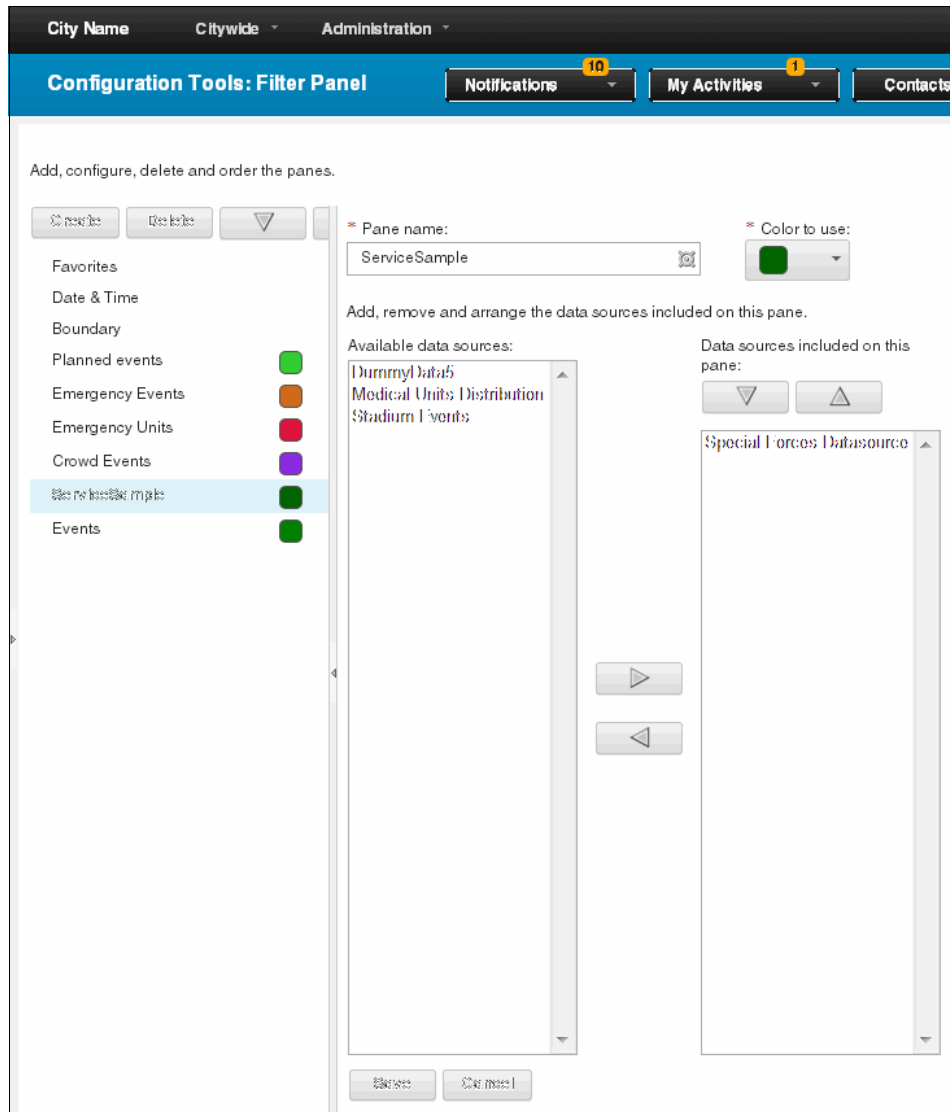


Figure 9-11 Filter pane for service URL data source

The data from this service is now visualized on the maps as shown in Figure 9-12.

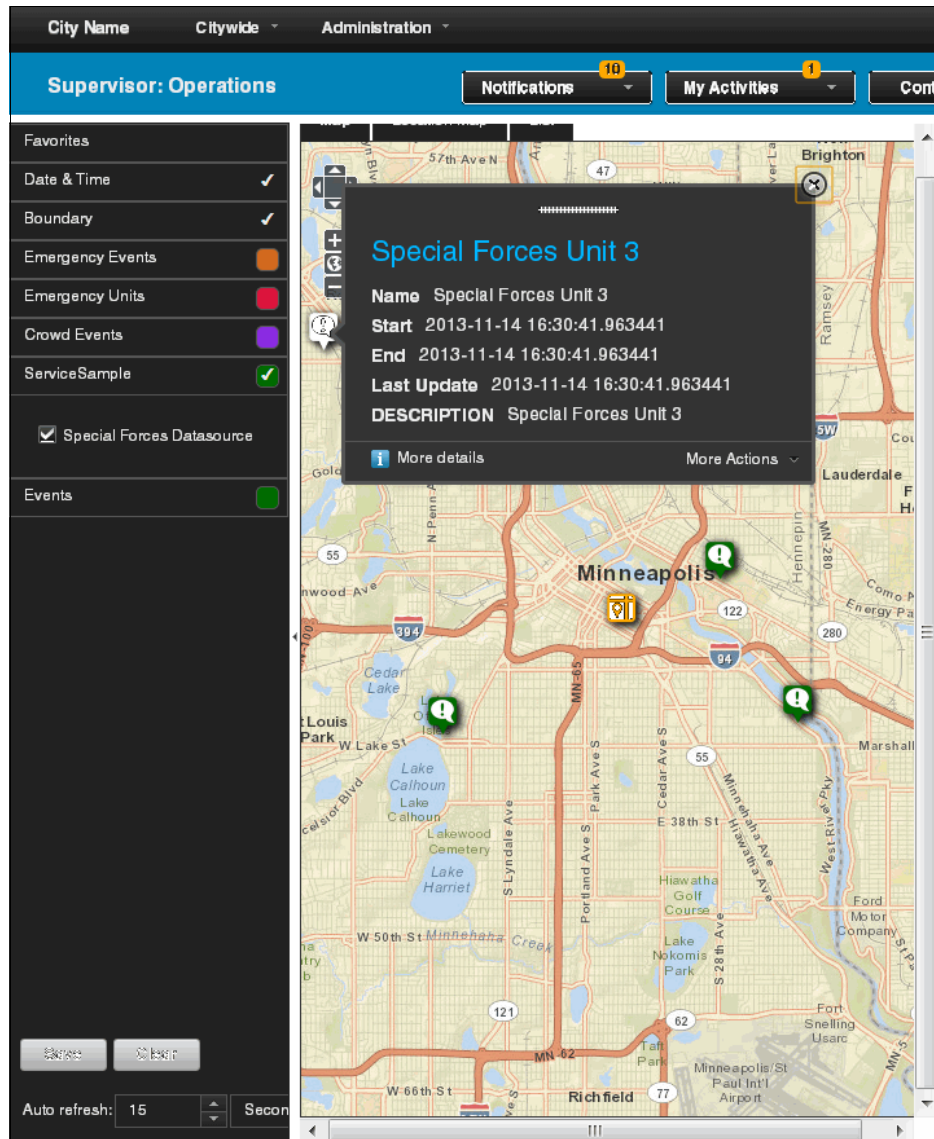


Figure 9-12 Special forces data map view (service URL)

The data from this service is now visualized on list views as shown in Figure 9-13 on page 347.

City Name Citywide Administration

Supervisor: Operations Notifications 10 My Activities 1 Contact

Map Location Map **List**

View data source Special Forces Datasource

Name	Start	End	Last Update	DESCRIPTION
Special Forces Unit 12	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	
Special Forces Unit 2	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	
Special Forces Unit 3	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	
Special Forces Unit 31	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	2013-11-14 16:30:41.96344	

Auto refresh: 15 Seconds

Figure 9-13 Special forces data List view (service URL)

9.6 Other options for integration

Consider the following options for integration with external systems:

- ▶ Java service APIs. Similar to REST services, server-side code for integration with external systems can be written to the IBM Intelligent Operations Center Java service API.
- ▶ Custom data source actions. They provide the capability to add new actions to the More Actions menu in the preview card for all associated data items. This action can provide the function to integrate with a third-party system. See 6.3, “Creating custom data source actions” on page 219.
- ▶ Preview card customization. Integration with a remote system can be implemented by creating a custom action on the preview card. See 6.5, “Customizing preview cards” on page 233.
- ▶ Custom portlets. Custom portlets can include custom code to integrate with external systems. See 6.6, “Creating custom portlets” on page 239.



Creating and viewing reports

Bringing data into the IBM Intelligent Operations Center solution is a critical process. Analyzing the data and displaying it in a format that enables city officials and business executives to make informed decisions are important to maximize the benefit of the solution.

This chapter describes how IBM Cognos Business Intelligence reports can be generated using data in IBM Intelligent Operations Center. Reports can help key stakeholders to understand business performance and historical trends. Reports can assist in decision making by providing more ways to visualize the data that is in your IBM Intelligent Operations Center system.

This chapter includes the following topics:

- ▶ 10.1, “Cognos reports versus Dojo charts” on page 350
- ▶ 10.2, “Use case overview” on page 350
- ▶ 10.3, “Setting up the development environment” on page 351
- ▶ 10.4, “Implementation summary” on page 352
- ▶ 10.5, “Implementation details” on page 353
- ▶ 10.6, “Deploying Cognos reports” on page 371
- ▶ 10.7, “Displaying Cognos reports” on page 371
- ▶ 10.8, “Problem determination” on page 372
- ▶ 10.9, “Additional considerations” on page 373

10.1 Cognos reports versus Dojo charts

IBM Intelligent Operations Center 1.6 includes support for Dojo charts as an alternative to creating Cognos reports in certain situations. Use Dojo charts as a simple and flexible tool to display important information and trends using data that spans multiple data sources. For more information about Dojo chart support in IBM Intelligent Operations Center, see 6.4, “Adding filter-driven result charts” on page 227.

Dojo charts in IBM Intelligent Operations Center 1.6 have the following characteristics:

- ▶ Filter-driven results. Charts are driven by the filter (charts react to the changes in filter configuration).
- ▶ Charts refresh instantly as a reaction to changes in filtering and data sources (refresh is data driven).
- ▶ They are only suitable for a moderate number of records, that is, in the hundreds.
- ▶ They require programming with JavaScript/Dojo topics.
- ▶ Data is retrieved by interacting with Dojo topics and Representational State Transfer (REST) APIs and restricted to minimal properties of the data source. The data can also be retrieved from the IOC Data Provider if the report is on the same page as the content viewer portlet and filter panel and it extends the IOC view or it is implemented as a widget in the content viewer.
- ▶ Complex filtering is not possible.

Cognos reports have the following characteristics:

- ▶ They do not require programming to adjust the generated model or create a custom chart.
- ▶ They require IBM Cognos Business Intelligence skills.
- ▶ They get data from data source database tables.
- ▶ Cognos reports can be used with or without other IBM Intelligent Operations Center portlets.
- ▶ They do not depend on the IBM Intelligent Operations Center REST APIs or Dojo topics.
- ▶ The refresh is by user request or by timeout. The refresh is not data driven.

Select the best charting option based on your data sources and requirements.

10.2 Use case overview

The administrators of City A want to generate reports to help them understand the type of first aid events happening over time. This information helps them to determine whether they have the correct operators allocated to manage the events. They also want to determine whether there is a correlation between the size of the crowd and the frequency of first aid events to assist in allocating resources for future planned events.

10.2.1 Actors

The actors are the city planners. These administrators are responsible for planning events in the city and allocating resources to support the events.

10.2.2 Desired outcome

Clear information available to administrators and presented in graphical report format for further analysis.

10.2.3 IBM Intelligent Operations Center programming artifacts

This use case incorporates the following programming artifacts:

- ▶ IBM Intelligent Operations Center data sources
- ▶ IBM Cognos Framework Manager models and Cognos reports

10.3 Setting up the development environment

When data is ingested into IBM Intelligent Operations Center 1.6, the data receiver can be set to automatically generate a Cognos Framework report model of that data source to facilitate report creation. *Models* provide a view of the data structure that is used to build reports. After the report models are generated, reports can be created using the IBM Cognos Business Intelligence Report Studio.

There is no automatic way to generate models that span multiple IBM Intelligent Operations Center data sources. The IBM Cognos Framework Manager is required to manually create report models that span multiple data sources.

IBM Cognos Framework Manager is included in the IBM Intelligent Operations Center Developer's Toolkit eAssembly and must be downloaded and installed separately on a Microsoft Windows workstation.

For information about downloading the IBM Intelligent Operations Center product images, including the IBM Intelligent Operations Center Developer's Toolkit eAssembly, see *Download IBM Intelligent Operations Center Version 1.6 image files* at this website:

<http://www-01.ibm.com/support/docview.wss?uid=swg24035524>

For information about installing IBM Cognos Framework Manager, see “Installing and configuring IBM Cognos Framework Manager” on page 380.

IBM Cognos Business Intelligence Report Studio is installed with IBM Intelligent Operations Center. You can access it through the IBM Intelligent Operations Center 1.6 administrative interface.

The IBM DB2 runtime client must be installed on the Microsoft Windows workstation to allow the IBM Cognos Framework Manager to connect to the data source.

You can download the IBM DB2 Data Studio, which includes the IBM DB2 runtime client, at no charge from this website:

<http://www-03.ibm.com/software/products/en/data-studio>

For more information about installing and using the IBM DB2 Data Studio, see the IBM DB2 Information Center:

<http://bit.ly/1nJ2LAi>

IBM Cognos Framework Manager and IBM Cognos Report Studio usage restrictions:
The license for IBM Cognos Framework Manager and IBM Cognos Report Studio included with IBM Cognos Business Intelligence enables you to create reports that use data from IBM Intelligent Operations Center data sources. These reports are displayed in IBM Intelligent Operations Center.

10.4 Implementation summary

The steps to generate and display a simple report using IBM Cognos Business Intelligence are summarized:

1. Verify the prerequisite configuration:
 - a. Ensure that the data source is configured with Report model set to Yes.
 - b. Inspect the imported data source to determine the table and column names to use in the report.
 - c. Ensure that IBM Intelligent Operations Center 1.6.0.1 is installed.
 - d. Install IBM DB2 Data Studio in the Microsoft Windows workstation where IBM Cognos Framework Manager is installed.
2. Determine the target table and column names.

To create reports on your data, you need to determine what target table names map to your data sources and what columns contain the data on which you want to report.
3. Create the simple report using the IBM Cognos Business Intelligence Report Studio.
4. Copy the IBM Intelligent Operations Center reports portlet and configure it to reference the new report.
5. Place the cloned IBM Intelligent Operations Center reports portlet on a page created with the IBM Intelligent Operations Center reporting theme.

The steps to generate and display a complex report spanning multiple data sources using IBM Cognos Business Intelligence are summarized:

1. Determine the target table and column names.

To create reports on your data, you need to determine what target table names map to your data sources and what columns contain the data on which you want to report.
2. Create a model spanning the data sources with IBM Cognos Framework Manager installed in a Microsoft Windows workstation.
3. Publish the model to the IBM Cognos Business Intelligence application components on the IBM Intelligent Operations Center analytics server.
4. Create the report using the IBM Cognos Business Intelligence Report Studio.
5. Copy the IBM Intelligent Operations Center reports portlet and configure it to reference the new report.
6. Place the cloned IBM Intelligent Operations Center reports portlet on a page created with the IBM Intelligent Operations Center reporting theme.

10.5 Implementation details

This section includes the step-by-step instructions for each step listed in the implementation summary.

10.5.1 Verify the prerequisite configuration

The following prerequisites are necessary to be able to create and display Cognos reports:

- Data sources are configured with Report model set to Yes.

Create the sample data sources, Emergency events and Crowd Gathering, as described in 3.2.4, “CSV data source” on page 55. On the data source configuration tool Basics tab, for Report model, select **Yes** as shown in Figure 10-1. The IBM Cognos Business Intelligence model is generated by the data receiver when the data from this data source is imported.

The screenshot shows the 'Create New Data Source: Step-by-Step Guide' wizard. At the top, there are eight numbered steps: 1. Acquire, 2. Basics (selected), 3. Minimal Properties, 4. Key & Full Properties, 5. Security, 6. Routing, 7. Actions, and 8. Appearance. Below the steps, there is a section titled 'Define some basic information about the data source. Depending on the options that you select, only the relevant fields are displayed.' This section is divided into two parts: 'Names and Description' and 'General'. In the 'Names and Description' part, there are two text input fields: 'Data source label' (containing 'Crowd Gathering') and 'Data source unique text identifier' (containing 'CRWD'). Below these is a 'Description' field containing 'Data source to collect crowd gathering information'. In the 'General' part, there are five dropdown menus, each with a question mark icon: 'Data for analysis' (No), 'Data for boundary' (No), 'Asset or resource data' (No), 'Data for correlation' (No), and 'Report model' (Yes). The 'Report model' dropdown is highlighted with a red rectangular box. At the bottom of the wizard, there are three buttons: 'Back', 'Next', and 'Cancel'.

Figure 10-1 Creating a data source with Report model Yes

- IBM Intelligent Operations Center Version 1.6.0.1 is installed.

To display Cognos reports, in IBM Intelligent Operations Center, Version 1.6.0.1 must be installed. This version provides a new theme to create reports pages and a new reports portlet for displaying Cognos reports. Reports pages are displayed as views in the IBM Intelligent Operations Center interface.

- ▶ IBM DB2 Data Studio must be installed.

To inspect the data source tables, you need access to an SQL query tool, such as the IBM DB2 Data Studio. IBM DB2 Data Studio is installed by default on the IBM Intelligent Operations Center 1.6 data server. IBM DB2 Data Studio can also be installed on a workstation as described in 10.3, “Setting up the development environment” on page 351.

Note: To start IBM DB2 Data Studio on the IBM Intelligent Operations Center data server, log on to the data server console as the root user and run the following commands:

```
xhost +
su - db2inst2
/opt/IBM/DS3.2.0/eclipse
```

Then, connect to the IOCDDB database and browse the database tables.

10.5.2 Determine the column names from the data source

When the IBM Intelligent Operations Center 1.6 data injection process imports data for a data source, several of the column names are not retained on the imported data. For example, in the Emergency Events data, the STARTDATETIME column is called STARTDATETIME. However, the column that contains the type of the event, named TYPE in the comma-separated value (CSV) file, has a column name of PROPERTY_5 in the table containing the imported data. In addition, the tables are not named after the data source name, that is, the incoming CSV file name or database table name, but instead are given a generic name, for example, IOC.TARGET_TABLE_97.

To create reports on your data, you need to determine what table names map to your data sources and what columns contain the data on which you want to report. You can get table and column information browsing the imported data source tables in the IBM Intelligent Operations Center 1.6 database using the IBM DB2 Data Studio tool.

Note: APAR PO02174 includes a fix for the column name issue. After the fix is installed, the column names reflect the names on any new data sources. For more information, see *IBM Intelligent Operations Center database and routing schema updates* at this website:

<http://www-01.ibm.com/support/docview.wss?uid=swg24036674>

You also need to determine the numeric identifier of the table that corresponds to the data source on which you want to report. Follow these steps to determine the numeric identifier:

1. Start IBM DB2 Data Studio and connect to the IBM Intelligent Operations Center IOCDDB database.
2. In the IBM DB2 Data Studio Tables view, locate the IOC.DATARECEIVER table, right-click it, and click **Data** → **Browse Data**.
3. Refer to the second column (JSON) to locate the data source unique name, which is the data source unique text identifier that you specified when you configured the data source. Note the corresponding DATASOURCEID and TARGET_TABLE_NAME columns.

Figure 10-2 on page 355 shows that the DATASOURCEID is 97 for the data source unique name EEVENTS6 and that the table containing the imported data is named IOC.TARGET_TABLE_97.

ID	JSON [CLOB(1048576)]	DATASOURCEID [INTEGER]	TARGET_TABLE_NAME [VA
82	{"uniqueName":"GRWD","colu	84	IOC.TARGET_TABLE_84
13	{"timeFormat":null,"updatable"	14	IOC.TARGET_TABLE_14
83	{"uniqueName":"EVENTS","colu	86	IOC.TARGET_TABLE_86
18	{"timeFormat":null,"updatable"	19	IOC.TARGET_TABLE_19
20	{"timeFormat":null,"updatable"	22	IOC.TARGET_TABLE_22
43	{"uniqueName":"DUMM5","colu	49	IOC.TARGET_TABLE_49
84	{"uniqueName":"PSP","columns	87	IOC.TARGET_TABLE_87
85	{"uniqueName":"CCARDS","colu	88	IOC.TARGET_TABLE_88
86	{"uniqueName":null,"columns":	89	IOC.TARGET_TABLE_89
93	{"uniqueName":"EEVENTS6","c	97	IOC.TARGET_TABLE_97

Figure 10-2 Locating the target table in IBM DB2 Data Studio

4. Close the IOC.DATARECEIVER data view and locate the target table.
5. Right-click the target table and select **Data** → **Browse Data**.
6. Inspect the data in the table to determine the columns of interest.

For example, in the target table `IOC.TARGET_TABLE_97`, the columns shown in Table 10-1 will be used to generate the report as described in 10.5.3, “Create the simple report” on page 355.

Table 10-1 Column names for the data source fields

IOC column name	CSV header	Description
STARTDATETIME	STARTDATETIME	The time stamp of the event
STARTDATE	Derived ^a	The date the event started
PROPERTY_5	TYPE	The type of event
PROPERTY_4	SEVERITY	Event severity
PROPERTY_3	SENDER	Who sent the event
PROPERTY_1	DESCRIPTION	Description

a. For the start date and end date time stamp columns, the data import process always creates separate columns in the data source table. There is one column for the original time stamp, the date, and the time. For example, in the data source table, there are three columns, `STARTDATETIME`, `STARTDATE`, and `STARTTIME`, and the data for all three is sourced from the single field `STARTDATETIME` in the CSV file.

10.5.3 Create the simple report

This section describes how to generate a report to graphically show the types of emergency events happening each day.

Note: You must use either Microsoft Internet Explorer or Mozilla Firefox to access the IBM Cognos Business Intelligence Report Studio. Report Studio is not supported on Google Chrome.

Follow these steps:

1. Log in to IBM Intelligent Operations Center portal.
2. Click **Administration** → **Solution Administration**.
3. Expand **System Administration** → **Administration Consoles**.
4. Under Analytics Server, click **Report administration** to launch IBM Cognos Connection.
5. In IBM Cognos Connection, click **Launch** → **Report Studio**, as shown in Figure 10-3.

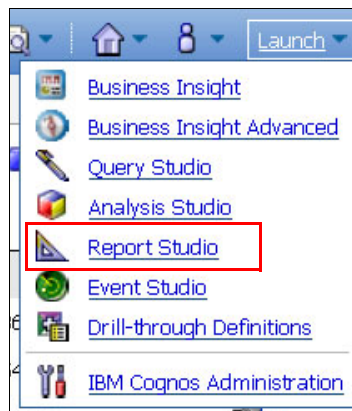


Figure 10-3 Launching Report Studio

6. In the Select a package (Navigate) window, click the **IOC** package.
7. In the IBM Cognos Report Studio Welcome window, click **Create new**.
8. Choose one of the following report types from the New pop-up window:
 - Blank
 - List
 - Crosstab
 - Chart
 - Map
 - Financial
 - Repeater Table
 - Blank Active Report
 - Active Report
 - Report Template
 - Existing

For example, to create a crosstab report, click **Crosstab** and click **OK**. The Report Studio window opens.

9. In the Source pane in the upper left of the window, expand the **IOC** package and expand the data source model with which you want to work. Figure 10-4 shows the Emergency Events model expanded, which lists, from the model, the columns on which you can report.

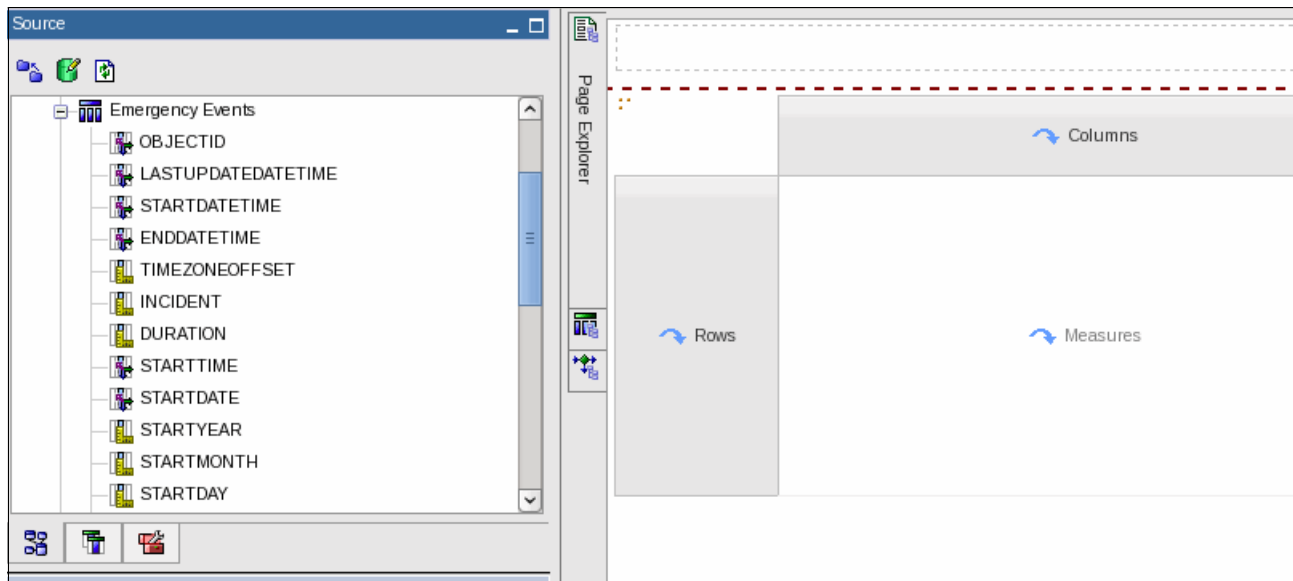


Figure 10-4 Expanded data source model and report design

10. The Property_5 column contains the TYPE column representing the type of event that occurred (see Table 10-1 on page 355). Drag **Property_5** from the Source pane and drop it on the Rows box in the Page Explorer pane.

11. The STARTDATE column is derived from the event time stamp and contains only the date the event occurred. Drag **STARTDATE** from the Source pane and drop it on the Columns box in the Page Explorer pane.

12. The example report counts the number of events raised per day. Any property can be used because the number of occurrences of that property will be counted instead of using the actual property data.

Drag **Property_4**, which corresponds to SEVERITY (see Table 10-1 on page 355), from the Source pane and drop it on the Measures box in the Page Explorer pane. Figure 10-5 shows the example report in the Page Explorer pane.



Figure 10-5 Report with columns selected

13. Follow these steps to configure the count of the measures:

- a. Click any of the <#1234#> text items to show the properties of that field in the Properties pane on the lower left of the Report Studio window.
- b. Scroll down to locate Data Item → Rollup Aggregate Function and change the value from None to **Count**.

14. Double-click **Double-click to edit text** and give the report a meaningful name.
15. Click **PROPERTY_4** in the Page Explorer window and locate Text Source → Source Type in the Properties pane as shown in Figure 10-6. Change the value to **Text**.

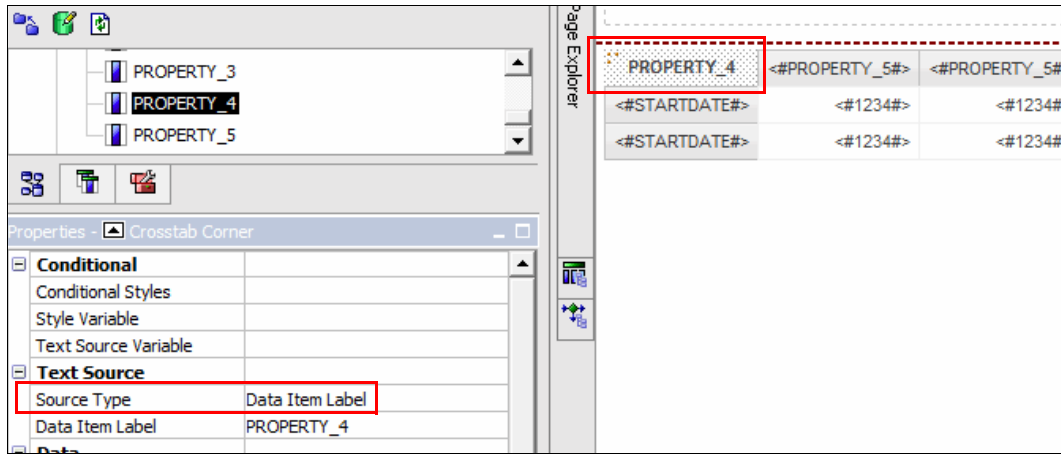


Figure 10-6 Changing the label type for PROPERTY_4

The PROPERTY_4 field in the Report Studio changes to read Double-click to edit text as shown in Figure 10-7. Double-click the link and change the text to a meaningful name.

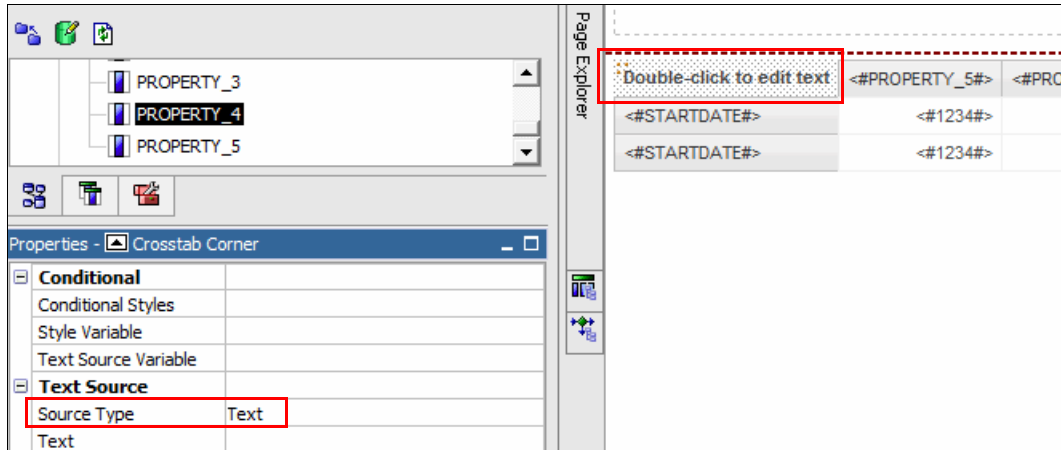


Figure 10-7 Changed Data Item Label to Text

16. Test the report by clicking the **Run Report** icon on the toolbar at the top of the Report Studio window as shown in Figure 10-8.



Figure 10-8 Run Report icon

A new window or tab opens in the browser and the report runs as shown in Figure 10-9 on page 359.

IBM Cognos Viewer										
Count of event types by day										
Event types	2013-11-18	2013-11-19	2013-11-20	2013-11-21	2013-11-22	2013-11-23	2013-11-24	2013-11-25	2013-11-26	2013-11-27
Car Accident	1	4	1	4	2	3	3	1		
Major Water Leak	3		2		3					2
Weapon Detected	1	1	6	2	1	1	2	3	2	
Man Fall Down		2	4	1	2	5		2	2	
Multi-vehicle Accident with Injuries			4	1			2			

Figure 10-9 Report results

17. Switch back to the Report Studio window and save the report.
18. Exit Report Studio.

10.5.4 Create a report across multiple data sources

To create a report that uses data from multiple IBM Intelligent Operations Center data sources, you first need to create a metadata model that models the data across multiple data sources. After the model is created, you can create the report.

Create the metadata model

To create the metadata model, start IBM Cognos Framework Manager on your Microsoft Windows workstation:

1. On IBM Cognos Framework Manager, click **Create a new project**.
2. On the New Project window, enter the Project Name and Location, clear the Use Dynamic Query Mode check box and click **OK**.
3. You might be prompted to accept the certificate from the analytics server. Accept the certificates, if required.
4. You will be prompted to log in to the IBM Intelligent Operations Center. On the Windows Security window, log on as the administrator user **wpsadmin**, and click **OK**.
5. On the Select Language pop-up window, choose your preferred language and click **OK** to launch the Metadata Wizard.
6. On the Metadata Wizard - Select Metadata Source window, click **Data Sources**, and click **Next**.
7. On the Metadata Wizard - Select Data Source window, click the **IOC** data source and click **Next**.
8. On the Metadata Wizard - Select Objects window, expand the **IOC** data source → expand **IOC** → expand **Tables**. Figure 10-10 on page 360 shows the expanded Select Objects window.

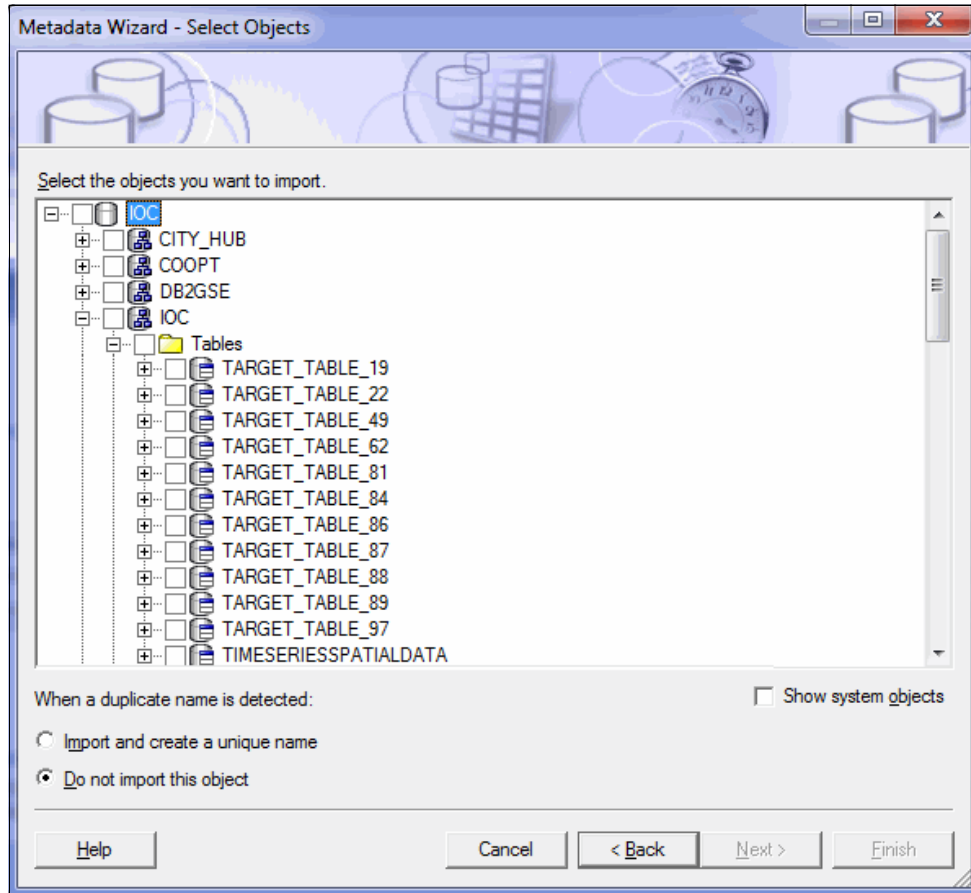


Figure 10-10 Metadata Wizard - Select Objects

Select the tables that you want to use in your model and click **Next**. For more information about determining which tables and columns to use in your report, see 10.5.2, “Determine the column names from the data source” on page 354.

In this example, TARGET_TABLE_84 and TARGET_TABLE_97 are used in the report.

9. On the Metadata Wizard - Generate Relationships window, choose the method that you want to use to generate relationships. The relationship type depends on the nature of your data. In the example, there are no specific primary keys or foreign keys that can be used to match the data and so matching is performed later manually.

Clear all relationship generating method check boxes and click **Import**. Figure 10-11 on page 361 shows the Generate Relationships window.

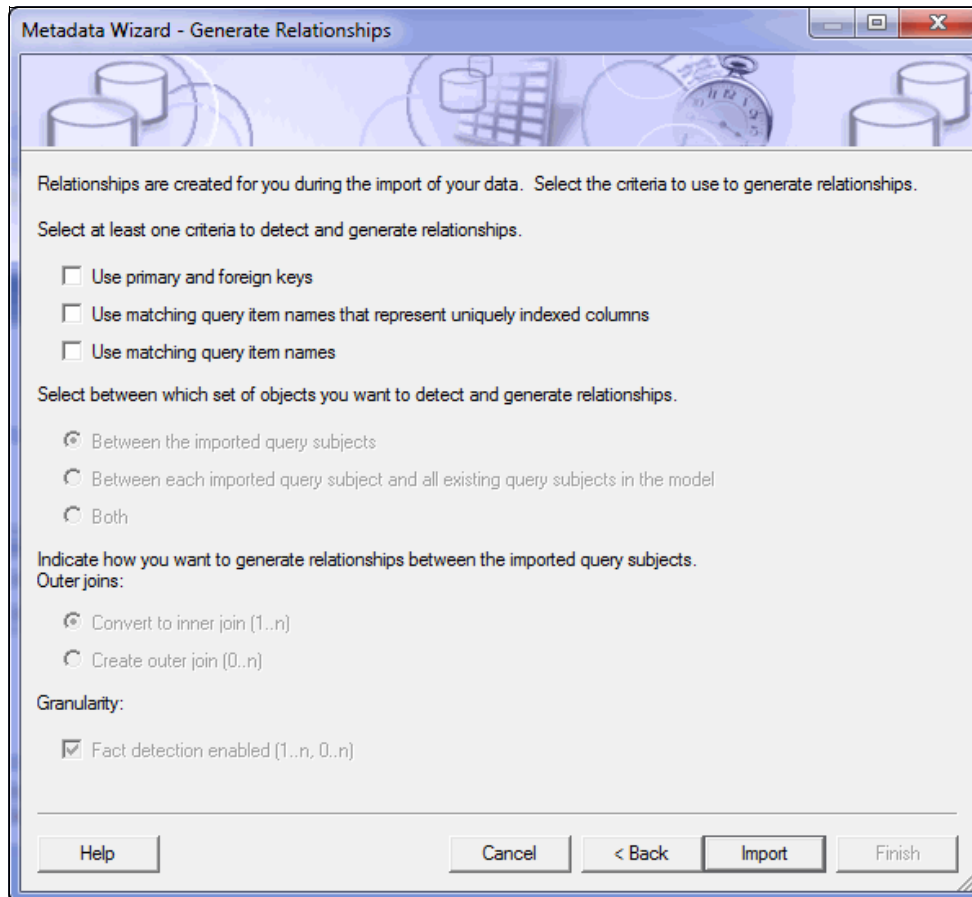


Figure 10-11 Metadata Wizard - Generate Relationships

10. On the Metadata Wizard - Finish window, click **Finish**. The metadata from the selected tables will be imported into the model.
11. On the IBM Cognos Framework Manager window, in the Project Viewer pane on the left, expand your project name and double-click **IOC**.
12. On the IBM Cognos Framework Manager window, click the **Actions** menu and select **Create → Relationship**.
13. On the Relationship Definition - New Relationship window, enter a name for the relationship.
14. On the Relationship Definition - New Relationship window, select the open folder icon next to the Query subject field on the left of the window and choose one of your imported tables. Click **OK**.
15. On the Relationship Definition - New Relationship window, select the open folder icon next to the Query subject field on the right of the window and choose the other imported table. Click **OK**.
16. On the Relationship Definition - New Relationship window, select **STARTDATE** from the list box on the left and also from the list box on the right. A link is created on the STARTDATE field between the two tables, as shown in Figure 10-12 on page 362.

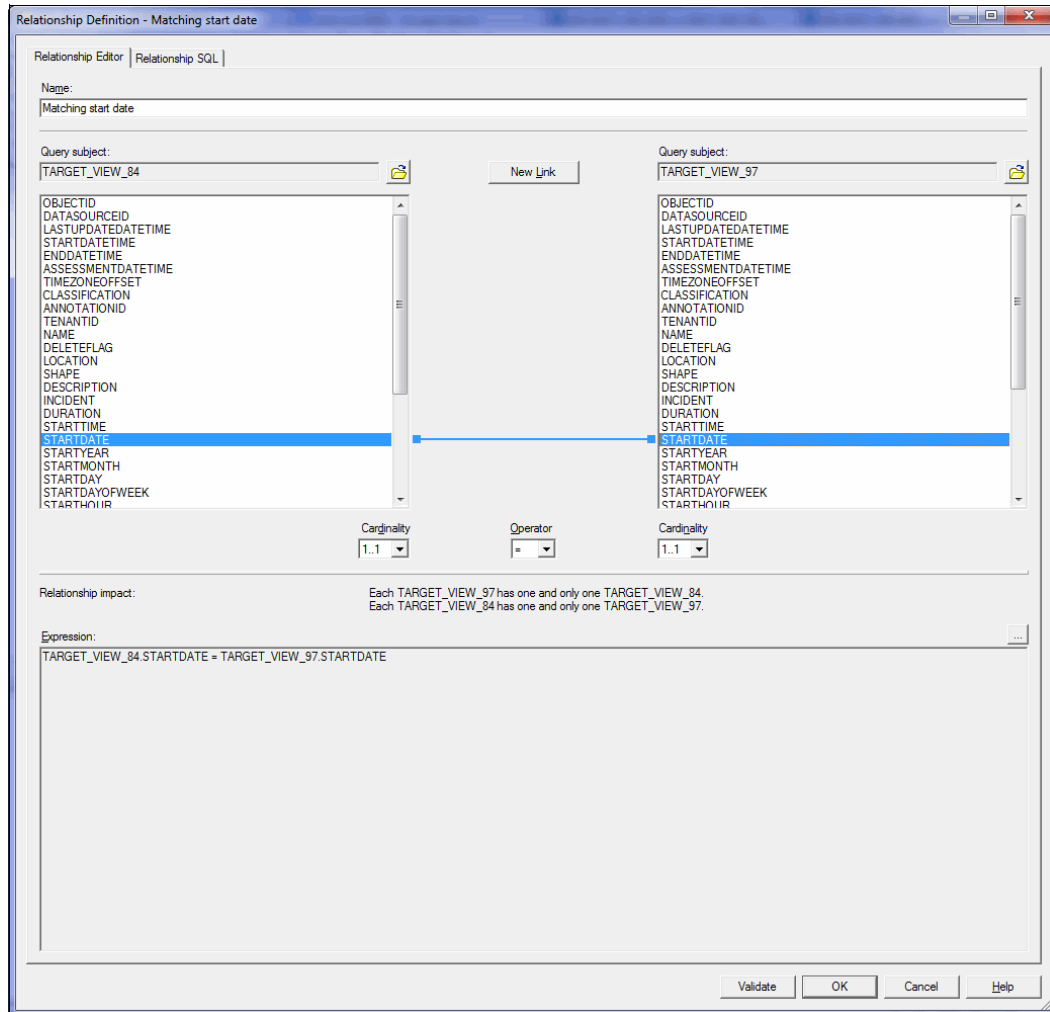


Figure 10-12 Relationship Definition - Matching start date

17. On the Relationship Definition - New Relationship window, click **Validate** to check that the relationship is valid. If there are no errors, click **OK**.

After creating the metadata model, the next step is to create a package containing the model and to publish the new package to the server so that a report can be created using the new model.

Follow these steps to create the package:

1. On the IBM Cognos Framework Manager window, select **Packages** in the Project Viewer pane.
2. Click the **Actions** menu and click **Create** → **Package**.
3. On the Create Package - Provide Name window, enter a name for the package and click **Next**.
4. On the Create Package - Define objects window, ensure that all objects are selected and click **Next**.
5. On the Create Package - Select Function Lists window, click **DB2** in the Selected function sets list box. Click the left arrow icon to move DB2 to the Available function sets list box as shown in Figure 10-13 on page 363. Click **Finish**.

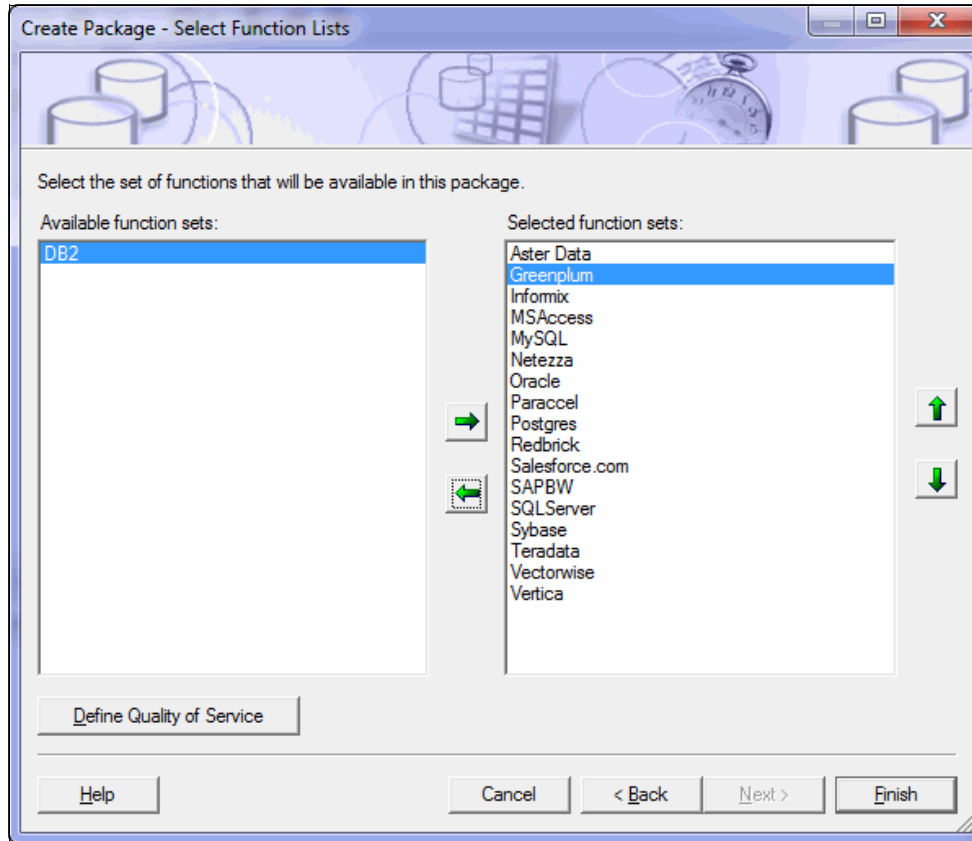


Figure 10-13 Create Package - Select Function Lists

6. On the IBM Cognos Framework Manager pop-up window, click **Yes** to start the Publish Package wizard.
7. On the Publish Wizard - Select Location Type, leave all settings as defaults as shown in Figure 10-14 on page 364, and click **Next**.

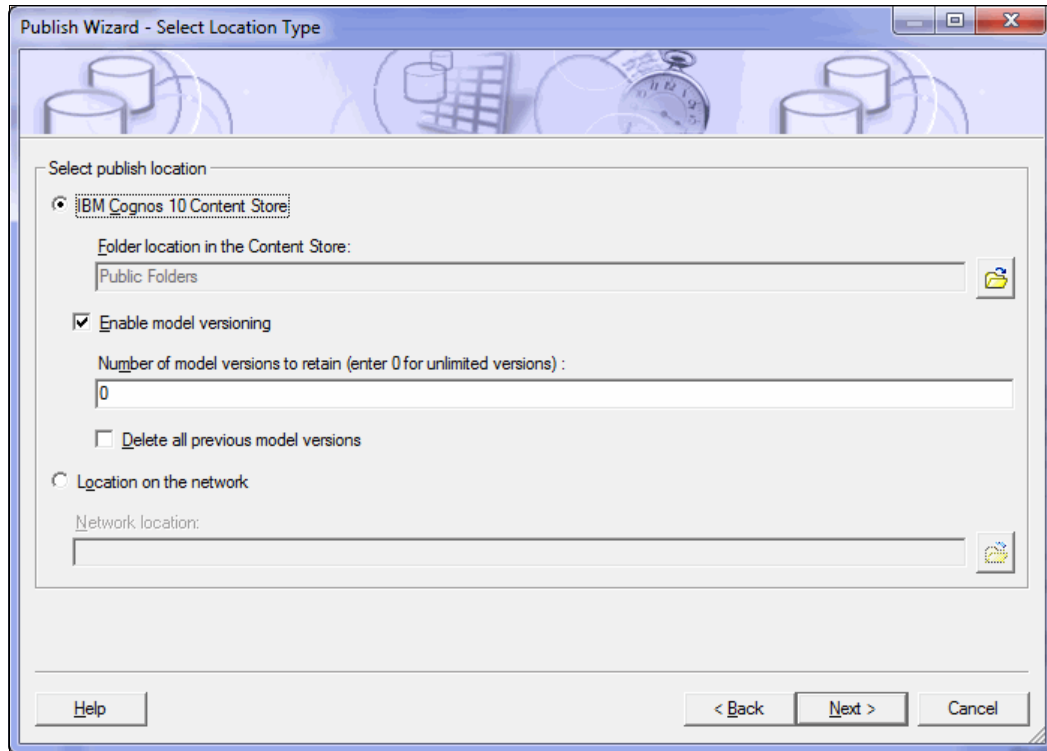


Figure 10-14 Publish Wizard - Select Location Type

8. On the Publish Wizard - Add Security window, click **Add**.
9. On the Select Users and Groups window, click **LDAP** in the Available Entries Directory list box. Then, click **GROUPS** to select the user groups that will be allowed to access the model in the Cognos Report Studio. Click the right arrow button to move the selected groups to the Selected Entries list box. Click **OK**.
10. The Publish Wizard - Add Security window is similar to Figure 10-15 on page 365. Click **Next**.

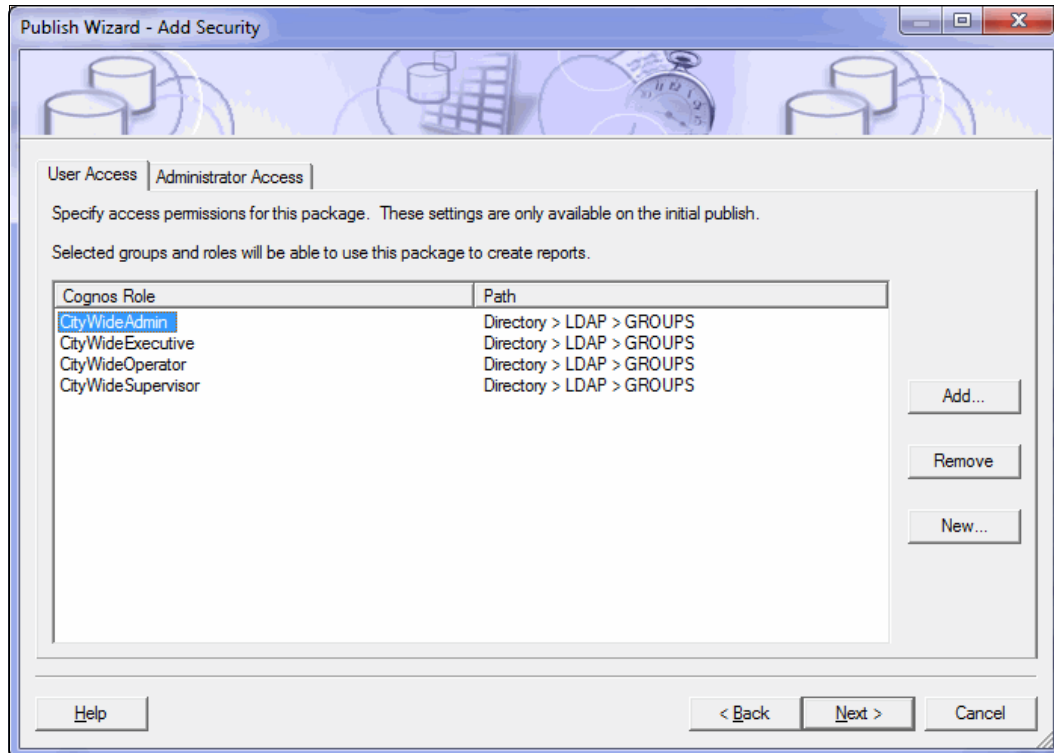


Figure 10-15 Publish Wizard - Add Security

11. On the Publish Wizard - Options window, click **Publish**.
12. On the Publish Wizard - Finish window, a message appears that states that you have successfully published the package to the following location: Public Folders → <your package name>. Click **Finish** to exit.

You can now create a report by using your new metadata model.

Creating the report

This section describes how to generate a graphical report showing the number of emergency events correlated with crowd size for each day.

Follow these steps:

1. Log in to IBM Intelligent Operations Center portal.
2. Click **Administration** → **Solution Administration**.
3. Expand **System Administration** → **Administration Consoles**.
4. Under Analytics Server, click **Report administration** to launch IBM Cognos Connection.
5. In IBM Cognos Connection, click **Launch** → **Report Studio**, as shown in Figure 10-16 on page 366.

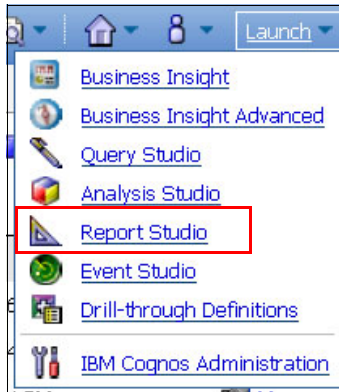


Figure 10-16 Launch Report Studio

6. On the Select a package (Navigate) pop-up window, click the package name that you created and published in “Create the metadata model” on page 359.
7. On the IBM Cognos Report Studio Welcome dialog box, click **Create new**.
8. Choose a report type from the New pop-up window:

- Blank
- List
- Crosstab
- Chart
- Map
- Financial
- Repeater Table
- Blank Active Report
- Active Report
- Report Template
- Existing

For example, to create a column chart, click **Chart**, and then click **OK**.

9. In the Insert Chart window, click **Column** in the left pane and choose one of the available column chart types from the right pane. Click **OK** to launch the Report Studio window.
10. On the left of the main Report Studio window, hover over the **Query Explorer** icon. The Query Explorer pop-up window opens (see Figure 10-17 on page 367).

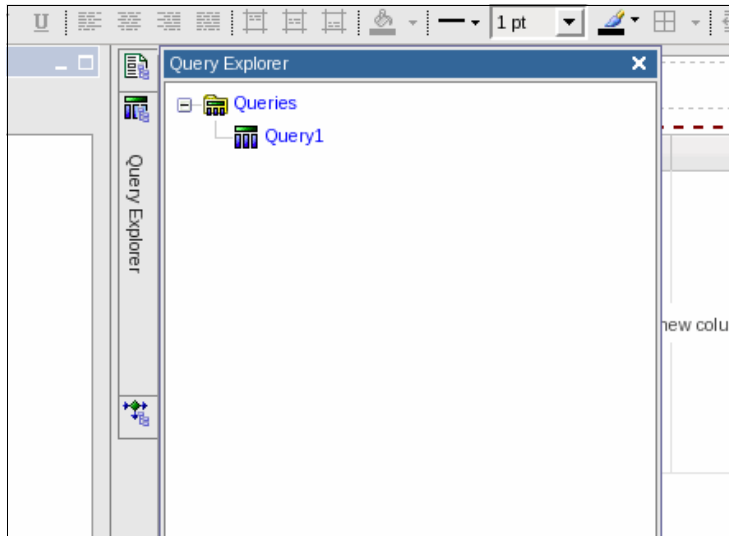


Figure 10-17 Open the Query Explorer

11. In the Query Explorer pop-up window, click **Queries**.

12. In the Toolbox pane on the left of the Report Studio, drag the **SQL** icon to the main report window and drop it on the **Query1** icon. The result is shown in Figure 10-18.

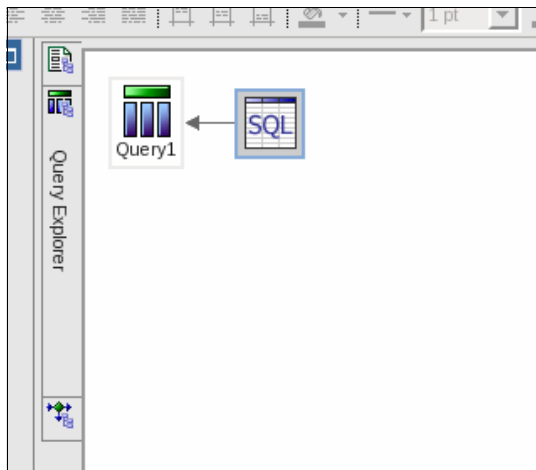


Figure 10-18 Query1 with SQL

13. In the properties window at the lower left of the Report Studio, double-click **Data** → **Data Source**.

14. In the Data Source pop-up window, click **IOC** and click **OK**. The selected data source appears as shown in Figure 10-19 on page 368.

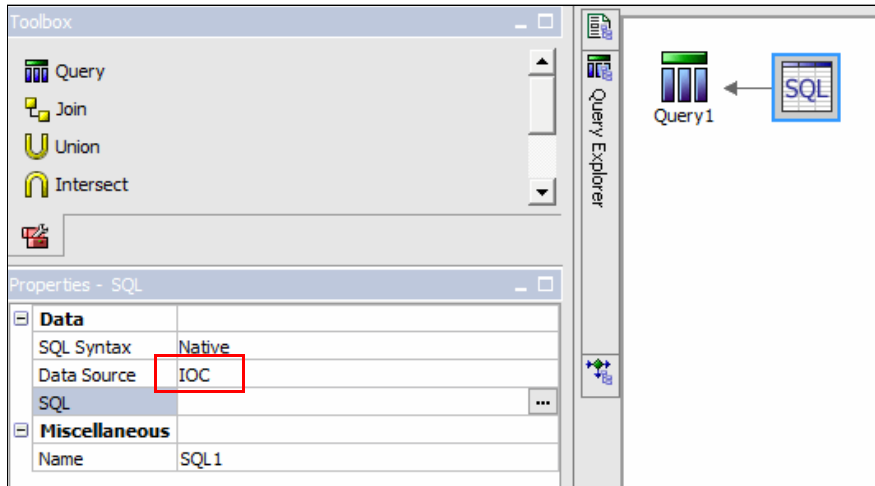


Figure 10-19 SQL query properties

15. In the Properties window at the lower left of the Report Studio, double-click **SQL**.

16. In the SQL pop-up window, copy the example SQL query from Figure 10-21 on page 369 and paste it into the SQL pop-up window as shown in Figure 10-20. Click **OK**.

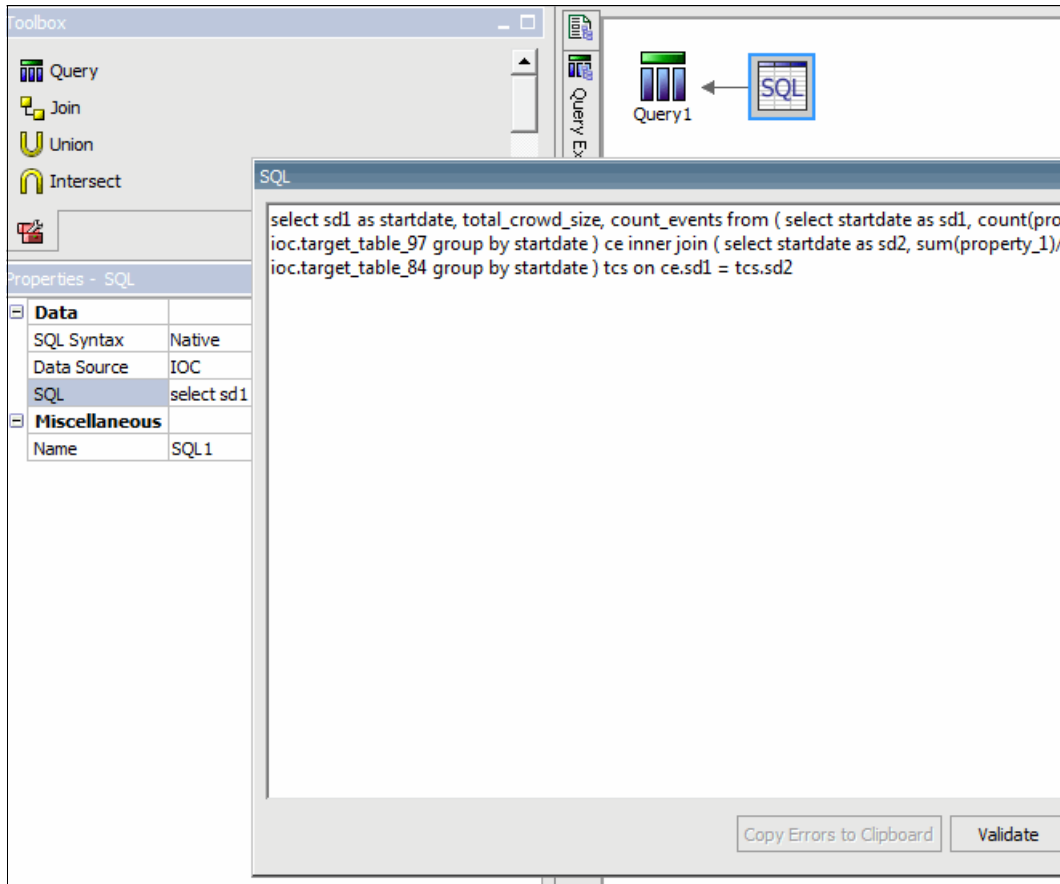


Figure 10-20 Pasting the SQL

```

select sd1 as startdate, total_crowd_size, count_events from ( select
startdate as sd1, count(property_2) as count_events from ioc.target_table_97
group by startdate ) ce inner join ( select startdate as sd2,
sum(property_1)/100 as total_crowd_size from ioc.target_table_84 group by
startdate ) tcs on ce.sd1 = tcs.sd2

```

Figure 10-21 Example SQL query

The SQL query joins the two tables in the data source on the STARTDATE field and aggregates the data. The events from TARGET_TABLE_97 are counted to get a count of events for each day. The crowd size from TARGET_TABLE_84 is added to get a total crowd size for each day and then divided by 100 to make the data more readable on the chart.

17. On the left of the main Report Studio window, hover over the **Page Explorer** icon. The Page Explorer pop-up window opens (see Figure 10-22).

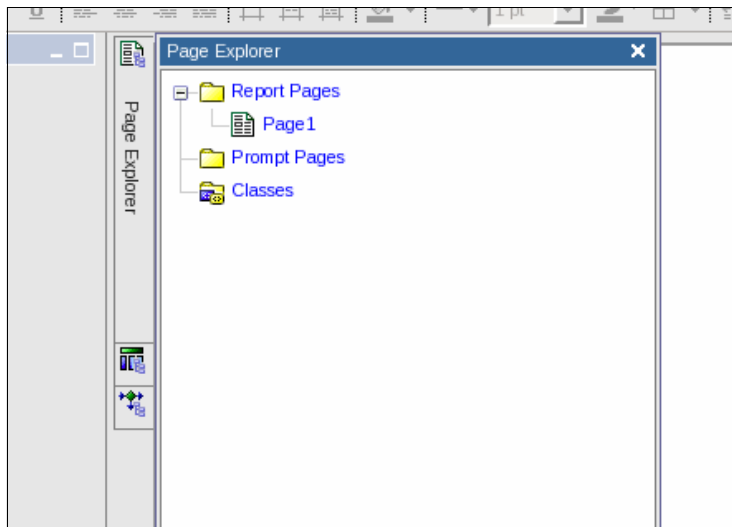


Figure 10-22 Open Page Explorer

18. On the Page Explorer pop-up window, click **Page1**.
19. In the upper left of the Report Studio at the bottom of the Source pane, click the **Data Items** icon to open the Data Items pane as shown in Figure 10-23 on page 370.

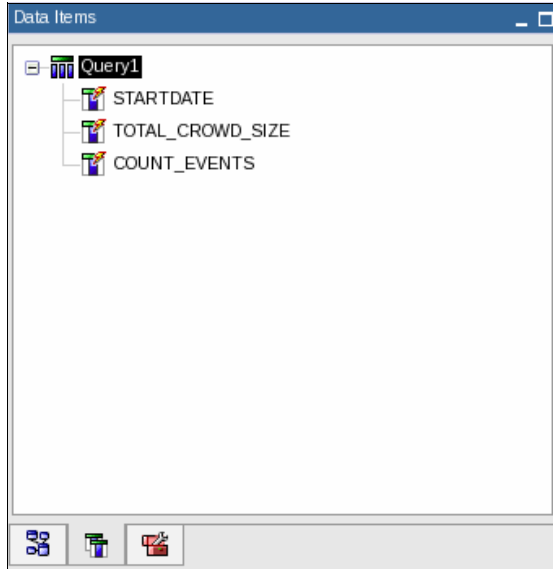


Figure 10-23 Data Items pane

20. In the Data Explorer pane, select the **STARTDATE** field and drag the field to the report in the main report window. Drop it on the box under Categories (x-axis).
21. In the Data Explorer pane, select both, **TOTAL_CROWD_SIZE** and **COUNT_EVENTS** and drag both fields to the box under Series (primary axis) in the main report window. Figure 10-24 shows an example.

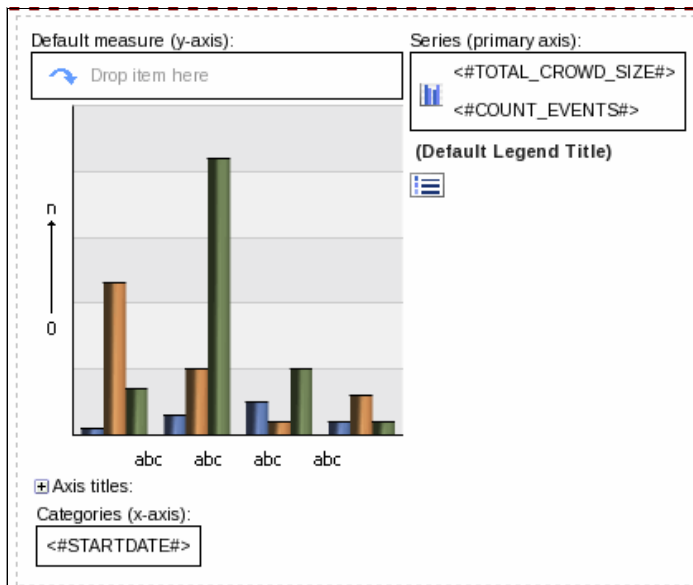


Figure 10-24 Report design

22. In the Report Studio window, click the field **TOTAL_CROWD_SIZE** under Series (primary axis). In the lower left of the Report Studio window, locate the Properties pane and click **Automatic** next to Aggregate Function. Change Automatic to **Total**.
23. In the main report window, click the field **COUNT_EVENTS** under Series (primary axis). On the lower left of the Report Studio window, locate the Properties pane and click **Automatic** next to Aggregate Function. Change Automatic to **Total**.

24. Double-click **Double-click to edit text** and give the report a meaningful name.

25. Test the report by clicking the **Run Report** icon on the toolbar at the top of the Report Studio window, as shown in Figure 10-8 on page 358.

A new window or tab opens in the browser, and the report runs as shown in Figure 10-25.

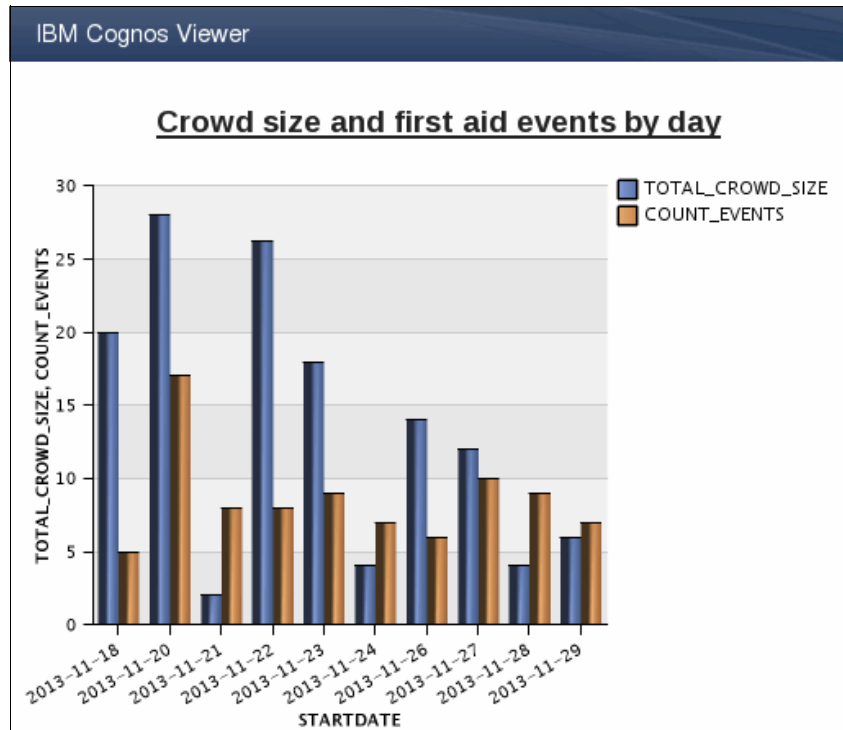


Figure 10-25 Running the custom model report

26. Switch back to the Report Studio window and save the report.

27. Exit Report Studio.

10.6 Deploying Cognos reports

Creating and saving the reports automatically deploy the reports to the Cognos server that is ready to be run. There is no further deployment that you need to do.

10.7 Displaying Cognos reports

To display the Cognos reports, you need the reports portlet `ioc_ui_reports_portlet` that is shipped with IBM Intelligent Operations Center 1.6.0.1.

Note: In IBM Intelligent Operations Center 1.6 (before the `ioc_ui_reports_portlet` was available), you were able to develop your own `iframe` portlet to show the Cognos reports.

Follow these steps to display IBM Cognos Business Intelligence reports:

1. Determine the default action URL for the reports.

For each report that you want to display, obtain the default action URL from the Cognos Connection console. Obtain the default action URL:

- a. Open the Cognos Connection console. In the Solution Administration view, click **System Administration** → **Administration Consoles** → **Report Administration**.
- b. In the Cognos Connection console, expand the **IOC** directory.
- c. In the reports portlet, locate the report that you want to display.
- d. In the actions column for the report, click **Set properties**.
- e. On the General tab, click **View the search path, ID, and URL**.
- f. Copy the URL from the Default Action URL entry.

Tip: You can improve the readability of your report when displayed in IBM Intelligent Operations Center by appending parameters to the URL that suppress the default headers and toolbar. Append the following line to the URL:

```
&cv.toolbar=false&cv.header=false
```

2. Create a copy of the report portlet `ioc_ui_reports_portlet` and configure it to display your reports using the modified default action URLs. A single instance of the report portlet can display up to three reports.
3. Create a page to display the report using the IBM Intelligent Operations Center 1.6.0.1 reports theme `IOC Portal 8 Reports Theme` and add the portlet to that page.

For detailed instructions about how to configure the reports portlet and the reports page, see *Displaying Cognos reports* at this website:

<http://bit.ly/1mIrnNb>

10.8 Problem determination

You might encounter some of the following common problems when building reports:

- ▶ Problems accessing the IBM Cognos Business Intelligence tools running on IBM Intelligent Operations Center:
 - Ensure that pop-up blockers allow pop-ups from the analytics server.
 - Ensure that access through firewalls to the analytics server is allowed.
- ▶ Problems understanding the data on which to build the report.

The most common problems encountered when building reports come from not fully understanding the data on which the reports are built. You might find it useful to query your data using SQL statements from a tool, such as IBM DB2 Data Studio. Using a tool helps you to ensure that the data on which you are reporting is what you think it is and that the data will support the type of queries that your report will be running.

For more information about Cognos reports problem determination, see the *Business Intelligence Troubleshooting Guide 10.2.0* at this website:

http://pic.dhe.ibm.com/infocenter/cbi/v10r2m0/nav/7_0

10.9 Additional considerations

The following considerations relate to preparing and displaying Cognos reports:

- ▶ Security considerations

When you import a data source into IBM Intelligent Operations Center, you assign security rights to that data source. The security rights apply to accessing the data through the IBM Intelligent Operations Center but do not have any bearing on the ability to create Cognos reports. That is, if you have access to the IBM Cognos Business Intelligence tools, you can access models for imported data sources.

When you create a report model and publish the model to the IBM Intelligent Operations Center analytics server, you assign security to the model. You must be a member of a group with at least read and execute access to the model to be able to generate and view reports on the modeled data.

- ▶ High Availability (HA) considerations

If you want to display Cognos reports in an HA environment, use the configuration that is described in *Configure the Cognos model in a high availability environment* at this website:

<http://bit.ly/1kY9aV6>

The host in the default action URL is the analytics server and the port is 9081. For example, the URL starts with: `http://<ioc_ana>:9081/ServletGateway`. If you want your report to be HA-enabled, you need to change the host name in the URL to the host name of the load balancer. For example, change the host name and port:

`https://<ioc_load_balancer>/ServletGateway`

This configuration ensures that the request to render the report is directed to an active cluster member by the load balancer.



A

Development tools

This appendix provides a summary of all the development tools used to develop the artifacts described in this book.

This appendix includes the following topics:

- ▶ “Development activities and tools” on page 376
- ▶ “Setting up IBM Rational Application Developer for WebSphere Software 8.5” on page 379
- ▶ “Installing and configuring IBM Cognos Framework Manager” on page 380

Development activities and tools

Table A-1 provides an index for the development activities described throughout this book and the corresponding development tools.

Table A-1 shows the following information:

- ▶ Development activity performed to extend the IBM Intelligent Operations Center solution.
- ▶ Development tool required.
- ▶ Whether the development tool is included with the IBM Intelligent Operations Center software.
- ▶ Whether the IBM Intelligent Operations Center 1.6 Premium User license enables you to use the tool for the specified development activity.
- ▶ Chapter and section in this book where setting up the development environment for the development activity is described in detail.

Notes: For more information, see these resources:

- ▶ For product license information, see *IBM Intelligent Operations Center Premium User V1.6 license* at this website:
<http://ibm.co/1aFSv3h>
- ▶ For information about downloading the IBM Intelligent Operations Center product images, including the IBM Intelligent Operations Center Developer's Toolkit eAssembly, see *Download IBM Intelligent Operations Center Version 1.6 image files* at this website:
<http://www-01.ibm.com/support/docview.wss?uid=swg24035524>

Table A-1 Development activities and corresponding tools

Activity	Development tool	Development tool included with IBM Intelligent Operations Center software	Usage license included with IBM Intelligent Operations Center premium user license	Chapter/Section
Create custom data source actions	IBM Rational Application Developer for WebSphere Software V8.5	No	No	6.3, "Creating custom data source actions" on page 219
	DB2 client application	Yes	Yes	
Customize the content viewer portlet	IBM Rational Application Developer for WebSphere Software V8.5	No	No	6.2, "Customizing the content panel" on page 197
Add filter-driven result charts	Text editor	No	No	6.4, "Adding filter-driven result charts" on page 227
Customize preview cards	IBM Rational Application Developer for WebSphere Software V8.5	No	No	6.5, "Customizing preview cards" on page 233
	DB2 client application	Yes	Yes	

Activity	Development tool	Development tool included with IBM Intelligent Operations Center software	Usage license included with IBM Intelligent Operations Center premium user license	Chapter/Section
Develop custom portlet	IBM Rational Application Developer for WebSphere Software V8.5	No	No	6.6, "Creating custom portlets" on page 239
Use the translation widget	IBM Rational Application Developer for WebSphere Software V8.5	No	No	6.6.4, "Using the translation widget and internationalization service" on page 248
Create and configure key performance indicators (KPIs)	IBM Business Monitor 8.0.1 with IBM Rational Application Developer 8.0.4 See "IBM Business Monitor and IBM Rational Application Developer special note" on page 378.	Yes Note: Included in the IBM Intelligent Operations Center V1.6 Developer's Toolkit eAssembly	Yes	4.3, "Setting up the development environment" on page 74
	IBM Integration Designer 8.0.1	No	No	
Manage IBM Intelligent Operations Center data sources	IBM Intelligent Operations Center data sources configuration tool	Yes	Yes	"Setting up the development environment" on page 53
Create and configure standard operating procedures (SOPs)	IBM Intelligent Operations Center standard operating procedures configuration tool	Yes	Yes	5.1.2, "Configuring SOP definitions" on page 115
Invoke an IBM SmartCloud Control Desk job plan	IBM Intelligent Operations Center standard operating procedures configuration tool	Yes	Yes	5.2.4, "Configuring automation activities" on page 132
	IBM SmartCloud Control Desk	Yes	Yes	
Develop mobile applications	IBM Rational Application Developer for WebSphere Software V8.5	No	No	8.3.1, "Setting up the development environment" on page 302
	IBM Worklight Consumer Edition See "IBM Worklight Consumer Edition special note" on page 378.	Yes	No	

Activity	Development tool	Development tool included with IBM Intelligent Operations Center software	Usage license included with IBM Intelligent Operations Center premium user license	Chapter/Section
Integrate with IBM SmartCloud Control Desk	IBM Intelligent Operations Center standard operating procedures configuration tool	Yes	Yes	9.2, "Integration with IBM SmartCloud Control Desk through SOP automation activity" on page 322
	IBM SmartCloud Control Desk	Yes	Yes	
Create Cognos models	IBM Cognos Framework Manager 10.2 See "IBM Cognos Framework Manager and IBM Cognos Report Studio special note" on page 379.	Yes Note: Included in the IBM Intelligent Operations Center V1.6 Developer's Toolkit eAssembly	Yes	10.3, "Setting up the development environment" on page 351
Create Cognos reports	IBM Cognos Report Studio included with IBM Cognos Business Intelligence 10.2 See "IBM Cognos Framework Manager and IBM Cognos Report Studio special note" on page 379.	Yes	Yes	10.3, "Setting up the development environment" on page 351

IBM Business Monitor and IBM Rational Application Developer special note

IBM Business Monitor V8.0.1.1 and IBM Rational Application Developer 8.0.4 are included with the IBM Intelligent Operations Center software. These products can only be used to configure KPIs to be used in the IBM Intelligent Operations Center platform.

IBM Worklight Consumer Edition special note

IBM Intelligent Operations Center 1.6 includes the IBM Worklight Consumer Edition V6.0 software product. The IBM Intelligent Operations Center Premium User V1.6 license entitles you to the following functions:

- ▶ Customize mobile applications delivered with IBM Intelligent Operations Center.
- ▶ Run mobile applications delivered with IBM Intelligent Operations Center as long as the *Prohibited Components* terms are met.

Prohibited Components: The licensee is not authorized to use any of the following components or functions:

- ▶ Change project structure (of IBM Worklight Consumer Edition)
- ▶ Process data that does not originate directly from the Principal Program (of IBM Worklight Consumer Edition) by the mobile application
- ▶ Create a new project (of IBM Worklight Consumer Edition)

The IBM Intelligent Operations Center Premium User V1.6 license does not entitle you to use the included IBM Worklight Consumer Edition software for the following purposes:

- ▶ Develop new mobile applications
- ▶ Run mobile applications (other than those included with IBM Intelligent Operations Center)

You need to purchase an IBM Worklight license separately to be entitled to develop and run mobile applications.

IBM Cognos Framework Manager and IBM Cognos Report Studio special note

The license for IBM Cognos Framework Manager and IBM Cognos Report Studio included with IBM Cognos Business Intelligence enables you to create reports that use data from IBM Intelligent Operations Center data sources. These reports are displayed in IBM Intelligent Operations Center.

Setting up IBM Rational Application Developer for WebSphere Software 8.5

Install IBM Rational Application Developer for WebSphere Software V8.5 and customize it for IBM Intelligent Operations Center development activities:

1. Follow the steps described in *Installing Rational Application Developer for WebSphere Software and optional software* at this website:

<http://bit.ly/11nQyRm>

2. During the installation, make sure that the following features are selected:

- Ajax, Dojo, and HTML
- Dojo Toolkit libraries
- XML
- WebSphere Portal Server 8.0 tools
- WebSphere Application Server 8.0 tools

3. After the installation is complete, follow these steps:

- a. Start IBM Rational Application Developer in a new workspace.
- b. Click **Window** → **Preferences** → **General** → **Capabilities**, and then, click **Enable All**.
- c. Click **OK** to finish.

Installing and configuring IBM Cognos Framework Manager

Download the IBM Cognos Framework Manager 10.2.0 Microsoft Windows Multilingual part number CID60ML. See *Download IBM Intelligent Operations Center Version 1.6 image files* at this website:

<http://www-01.ibm.com/support/docview.wss?uid=swg24035524>

Run the Framework Manager installer and complete the installation using the default configuration.

The workstation also needs the IBM DB2 runtime client installed to allow the Framework Manager to connect to the data source. You can download the IBM DB2 Data Studio, which includes the IBM DB2 runtime client, at no charge from IBM:

<http://www-03.ibm.com/software/products/en/data-studio>

For more information about installing and using the IBM DB2 Data Studio, see the IBM DB2 Information Center:

<http://bit.ly/1nJ2Lai>

Note: You must be able to access the IBM Intelligent Operations Center 1.6 Analytics Server from your workstation.

After the installation is complete, run the IBM Cognos Configuration tool. Configure IBM Cognos Business Intelligence to connect to your IBM Intelligent Operations Center Analytics Server.

On the IBM Cognos Configuration window, click **Environment** in the navigation pane on the left and set the following settings:

- ▶ Gateway URI = `https://<ioc_ana>:9444/ServletGateway/servlet/Gateway`
- ▶ Dispatcher URI for external applications = `http://<ioc_ana>:9082/p2pd/servlet/dispatch/ext`

Leave all other settings at their default values and click **File** → **Save** from the menu. Figure 10-26 on page 381 shows an example of the configuration.

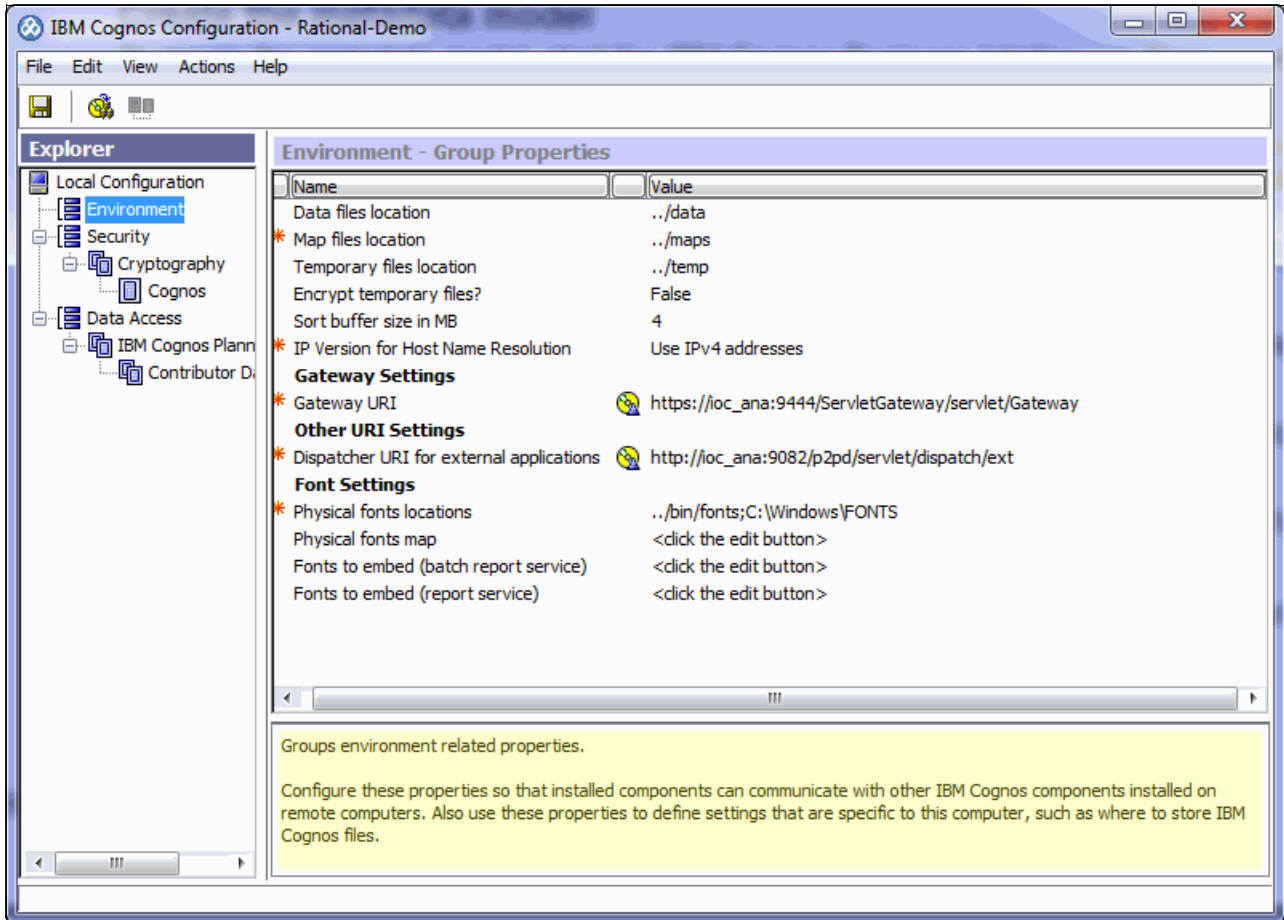


Figure 10-26 IBM Cognos Configuration

For more information about installing and configuring Framework Manager, see the IBM Cognos Business Intelligence Information Center:

<http://bit.ly/1nrN96X>



B

Samples provided with this book

This appendix includes the summary information for the samples provided with this book.

This chapter includes instructions to import the sample projects for each chapter into the IBM Rational Application Developer workspace.

The IBM Intelligent Operations Center development libraries are needed in the development environment to compile custom portlet code. This chapter describes how to obtain and install the development libraries.

For information about downloading the samples, see Appendix C, “Additional material” on page 397.

Disclaimer: IBM does not warrant or represent that the code provided is complete or up-to-date. IBM does not warrant, represent, or imply reliability, serviceability, or function of the code. IBM is under no obligation to update content or provide further support.

All code is provided “as is,” with no warranties or guarantees whatsoever. IBM expressly disclaims to the fullest extent permitted by law all express, implied, statutory, and other warranties, guarantees, or representations, including, without limitation, the warranties of merchantability, fitness for a particular purpose, and non-infringement of proprietary and intellectual property rights.

You understand and agree that you use these materials, information, products, software, programs, and services, at your own discretion and risk and that you will be solely responsible for any damages that may result, including loss of data or damage to your computer system.

In no event will IBM be liable to any party for any direct, indirect, incidental, special, exemplary or consequential damages of any type whatsoever related to or arising from use of the code found herein, without limitation, any lost profits, business interruption, lost savings, loss of programs or other data, even if IBM is expressly advised of the possibility of such damages.

This exclusion and waiver of liability applies to all causes of action, whether based on contract, warranty, tort, or any other legal theories.

Extracting the samples compressed file

Extract the sample projects included with this book:

1. Download the SG248201.zip compressed file on your desktop as described in Appendix C, “Additional material” on page 397.
2. Extract the SG248201.zip compressed file. You have a directory structure as shown in Figure B-1.

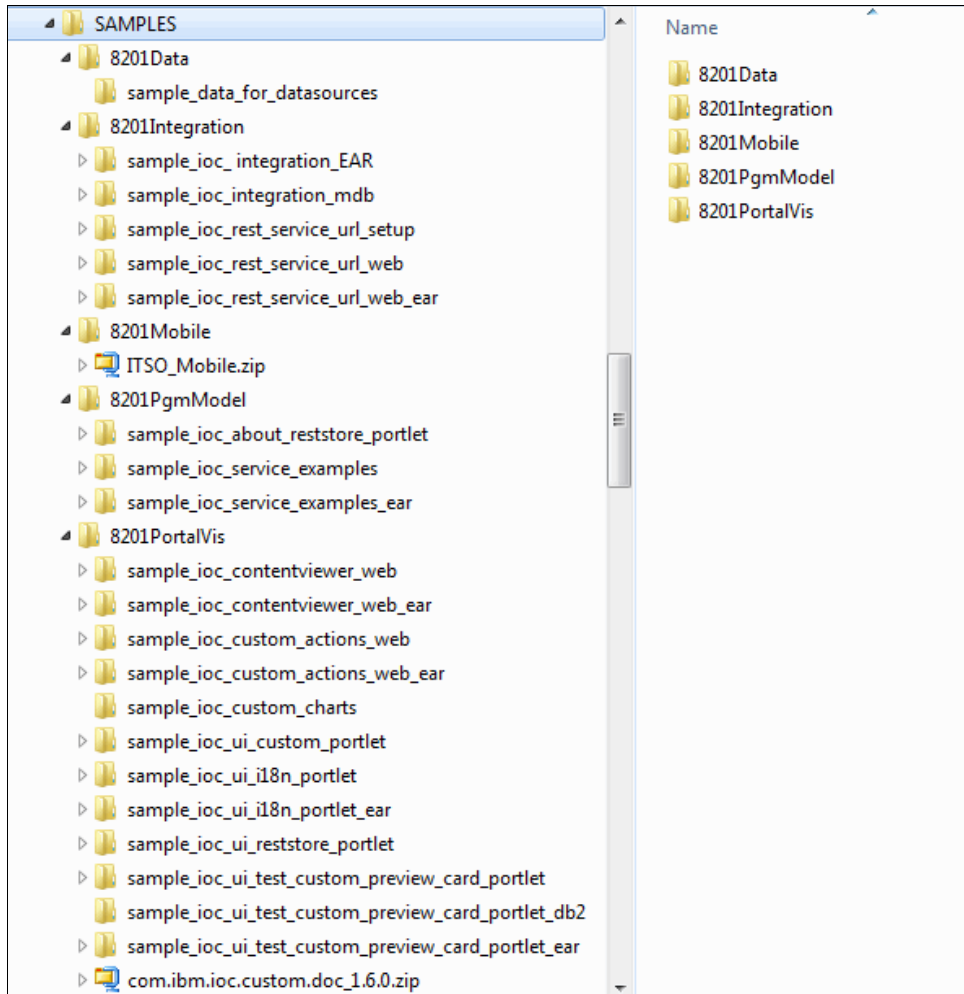


Figure B-1 SAMPLES directory included with this book

Table B-1 on page 386 lists the sample directories and the corresponding chapters in this book.

Table B-1 Sample directories provided with this book

Sample directory	Chapter	Description
8201Data	Chapter 3, "Integrating with data sources" on page 45	See "Integrating with data sources samples" on page 386.
8201Integration	Chapter 9, "Integrating with external systems" on page 321	See "Integrating with external systems samples" on page 387.
8201Mobile	Chapter 8, "Developing mobile applications" on page 299	See "Developing mobile application samples" on page 387.
8201PgmModel	Chapter 1, "IBM Intelligent Operations Center programming model overview" on page 1	See "IBM Intelligent Operations Center programming model overview samples" on page 389.
8201PortalVis	Chapter 6, "Customizing the operations display" on page 191	See "Customizing the operations display samples" on page 389.

Integrating with data sources samples

There is one sample described in Chapter 3, "Integrating with data sources" on page 45 and that is the sample data for data sources.

Table B-2 lists the sample, corresponding directory, and description.

Table B-2 Integrating with data sources: Samples and directories

Sample	Directory name	Description
Sample data for data sources	sample_data_for_datasources	<p>Contains sample files used in data source creation in the video tutorials and other chapters in the book:</p> <ul style="list-style-type: none"> ▶ CrowdGathering.csv: Sample file for the comma-separated value (CSV) data source. Used in the video tutorial <i>Creating a CSV data source</i> at this website: http://youtu.be/4fnr7w-hDE. Key performance indicators (KPIs) also use this case. ▶ FirstAidEvents.csv: Sample file for CSV data source used to create custom Cognos report and mobile application. ▶ EmergencyUnits.csv: Sample file for CSV data source used to show events in custom mobile application. ▶ emergency_unit_alert.xml: Sample Common Alerting Protocol (CAP) message that is used to test CAP data source in video tutorial <i>Creating a CAP data source</i> at this website: http://youtu.be/00mQ8phuSsqo.

Integrating with external systems samples

There are two sample projects described in Chapter 9, “Integrating with external systems” on page 321:

- ▶ Integration topic message-driven bean (MDB) sample project
- ▶ Representational State Transfer (REST) Service URL sample project

Table B-3 lists the samples, corresponding projects, and descriptions.

Table B-3 Integrating with external systems chapter: samples and projects

Sample	Project	Description
Integration topic Message-Driven Bean	sample_ioc_integration_mdb	Message-driven Bean Project
	sample_ioc_integration_EAR	Associated Enterprise Application Project
REST Service URL	sample_ioc_rest_service_url_setup	Database scripts and JavaScript Object Notation (JSON) object needed to configure the sample: The SAMPLE_SERVICE_URL_DATA script files define the sample table and records returned by the sample REST service. The I18nResource.sql script defines the i18n resources used by the service URL data source. The datasource.json contains the JSON object to be sent with the datasource-service POST method to create the data source. The service property URL in the JSON object needs to be updated to address the sample REST service through the IBM Intelligent Operations Center 1.6.0.1 web server (no port number) "service": "https://<ioc_ibs>/sample/custom/service/url/sample-service/features"
	sample_ioc_rest_service_url_web	The web project that provides the REST service
	sample_ioc_rest_service_url_web_ear	The associated enterprise application project

For instructions to import the samples into the IBM Rational Application Developer workspace, see “Key considerations for using Rational Application Developer” on page 391 and “Importing the sample projects into IBM Rational Application Developer workspace” on page 392.

Developing mobile application samples

There is one sample described in Chapter 8, “Developing mobile applications” on page 299, which shows how to develop an end-to-end mobile application that consumes REST services from IBM Intelligent Operations Center.

Table B-4 on page 388 lists the samples, corresponding projects, and descriptions.

Table B-4 Developing mobile applications chapter: Samples and projects

Sample	Project	Description
Mobile application	ITSOMobile (extracted from ITSOMobile.zip sample file)	IBM Worklight project that includes all the artifacts for the mobile app sample. The sample shows how a mobile application can be built to communicate with the IBM Intelligent Operations Center REST service.

Importing the sample projects into IBM Worklight Studio workspace

Import the sample projects into the IBM Worklight Studio workspace:

1. Locate the subdirectory \SAMPLES\8201Mobile, which was created after you extracted the SG248201.zip file, as described in “Extracting the samples compressed file” on page 385.
2. Open the IBM Worklight Studio workspace, and click **File** → **Import** → **Existing Projects into Workspace** → **Next**.
3. In the next window, click **Select archive file**, and then, click **Browse**. Select the archive file **ITSOMobile.zip**, and then, click **Finish**, as shown in Figure B-2.

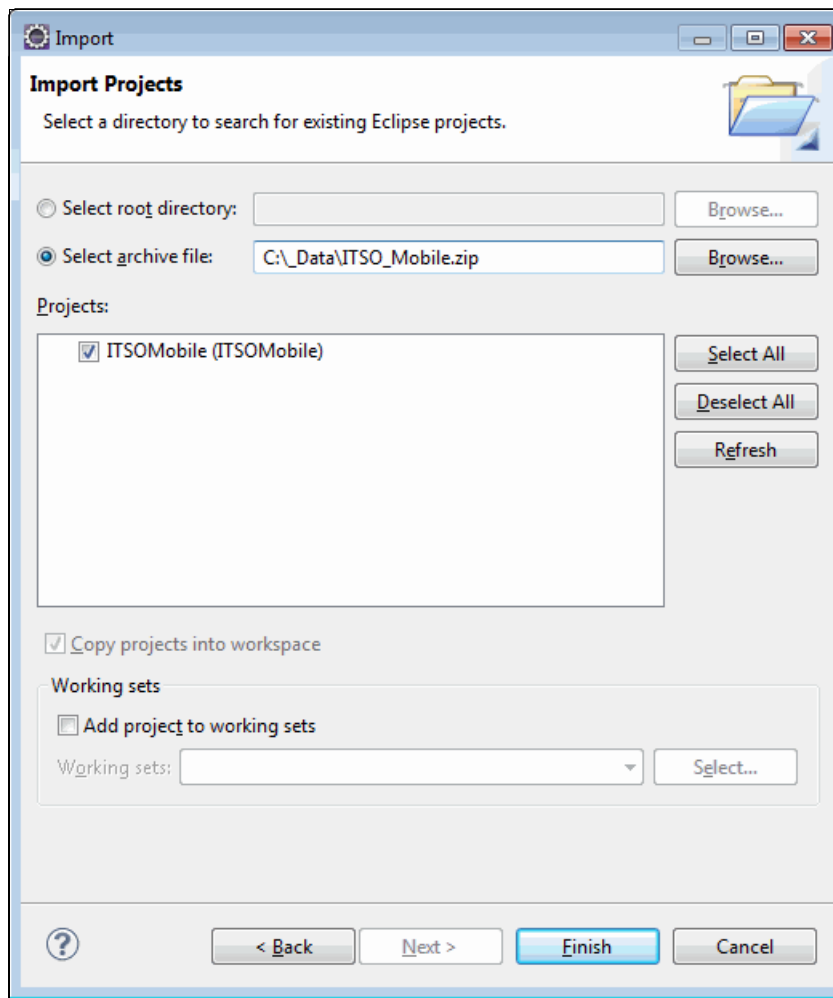


Figure B-2 Importing the sample IBM Worklight project

IBM Intelligent Operations Center programming model overview samples

There are two sample projects described in Chapter 1, “IBM Intelligent Operations Center programming model overview” on page 1:

- ▶ Client-side service invocation sample
- ▶ Server-side service invocation sample

Table B-5 lists the samples, corresponding projects, and descriptions.

Table B-5 IBM Intelligent Operations Center programming model overview chapter: Samples and projects

Sample	Project	Description
Client-side service invocation	sample_ioc_about_reststore_portlet	The web project that contains the portlet that prints the list of installed products in IBM Intelligent Operations Center using the REST service invocation
Server-side service invocation	sample_ioc_service_examples	The web project that contains the Java code to invoke the server-side data service to print the list of installed products in IBM Intelligent Operations Center
	sample_ioc_service_examples_ear	The enterprise application project that contains the web project with the server-side code

For instructions to set up the development environment and import the samples into the IBM Rational Application Developer workspace, see “Key considerations for using Rational Application Developer” on page 391, “Importing the sample projects into IBM Rational Application Developer workspace” on page 392, and “Installing IBM Intelligent Operations Center JARS in the development environment” on page 395.

Customizing the operations display samples

There are eight samples described in Chapter 6, “Customizing the operations display” on page 191:

- ▶ Custom content viewer portlet tab sample
- ▶ Custom data source actions sample
- ▶ Custom filter-driven result charts sample
- ▶ Custom preview card sample
- ▶ Custom generic portlet sample
- ▶ Internationalization service widget sample
- ▶ Eclipse plug-in with custom help topic sample
- ▶ Update filter pane colors with RestStore portlet

Table B-6 on page 390 lists the samples, corresponding directory, and descriptions.

Table B-6 Customizing the operations display: Samples and directories

Sample	Directory name	Description
Custom content viewer portlet tab	sample_ioc_contentviewer_web	Rational Application Developer project that contains sample code for custom content viewer portlet tab, sample servlet, and widget code.
	sample_ioc_contentviewer_web_ear	The associated enterprise application project.
Custom data source actions	sample_ioc_custom_actions_web	Rational Application Developer project that contains sample code for the servlet that implements service for the data source actions.
	sample_ioc_custom_actions_web_ear	The associated enterprise application project.
Custom filter-driven result charts	sample_ioc_custom_charts	Sample contains source code for two artifacts: <ul style="list-style-type: none"> ▶ ChartingPortletView.jsp Example of standard charting portlet customization. ▶ CustomChart.js Sample chart widget.
Custom preview card	sample_ioc_ui_test_custom_preview_card_portlet	Rational Application Developer project that contains source code for custom preview card servlet and widget.
	sample_ioc_ui_test_custom_preview_card_portlet_db2	Contains DB2 scripts to create and populate the table for the sample data source that must be configured to use custom preview cards. It also contains the INSERT statement to configure the custom preview cards for this data source.
	sample_ioc_ui_test_custom_preview_card_portlet_ear	The associated enterprise application project.
Custom generic portlet	sample_ioc_ui_custom_portlet	Rational Application Developer project with the sample code for the generic <i>Hello World</i> portlet for IBM Intelligent Operations Center.
Internationalization service widget	sample_ioc_ui_i18n_portlet	Rational Application Developer project that contains sample code for the portlet that uses the internationalization (i18n) widget.
	sample_ioc_ui_i18n_portlet_ear	The associated enterprise application project.

Sample	Directory name	Description
Eclipse plug-in with custom help topic	File name: com.ibm.ioc.custom.doc_1.6.0.zip	Contains the sample Eclipse plug-in with custom help topic.
Update filter pane colors with RestStore portlet	sample_ioc_ui_reststore_portlet	Rational Application Developer project with the sample code for the reststore portlet. The sample shows how to create a custom portlet that uses RestStore to display the existing filter panes and listen to events for changes in the filter panes, and refresh the view.

For instructions to set up the development environment and import the samples into the IBM Rational Application Developer workspace, see “Key considerations for using Rational Application Developer” on page 391, “Importing the sample projects into IBM Rational Application Developer workspace” on page 392, and “Installing IBM Intelligent Operations Center JARS in the development environment” on page 395.

Key considerations for using Rational Application Developer

There are three main points to remember before you start to develop extensions and customizations for IBM Intelligent Operations Center 1.6 with IBM Rational Application Developer:

- ▶ **Version considerations.** IBM Intelligent Operations Center contains a number of different components; use Rational Application Developer Version 8.5.1 to develop custom applications for portal and application server components.
- ▶ **Licensing restrictions.** IBM Rational Application Developer 8.0.4 is included with the IBM Intelligent Operations Center software. This product, in conjunction with IBM Business Monitor V8.0.1.1, which is also provided with IBM Intelligent Operations Center software, can only be used to configure KPIs for use in the IBM Intelligent Operations Center platform. It cannot be used for portal or application server development. Licenses for IBM Rational Application Developer 8.5.1 must be purchased separately to develop portal and application server applications for IBM Intelligent Operations Center Version 1.6.
- ▶ **Runtime environments.** To install and configure IBM Rational Application Developer 8.5.1, follow the standard process described in *Installing Rational Application Developer for WebSphere Software and optional software* at this website:

<http://bit.ly/1jkrDtC>

Make sure that the following runtime environments are checked for deployment as shown in Figure B-3 on page 392:

- WebSphere Application Server 8.5, which is required for the sample sample_ioc_service_examples from 1.5.2, “Server-side service invocation sample” on page 19
- WebSphere Application Server 8.0
- WebSphere Portal Server 8.0

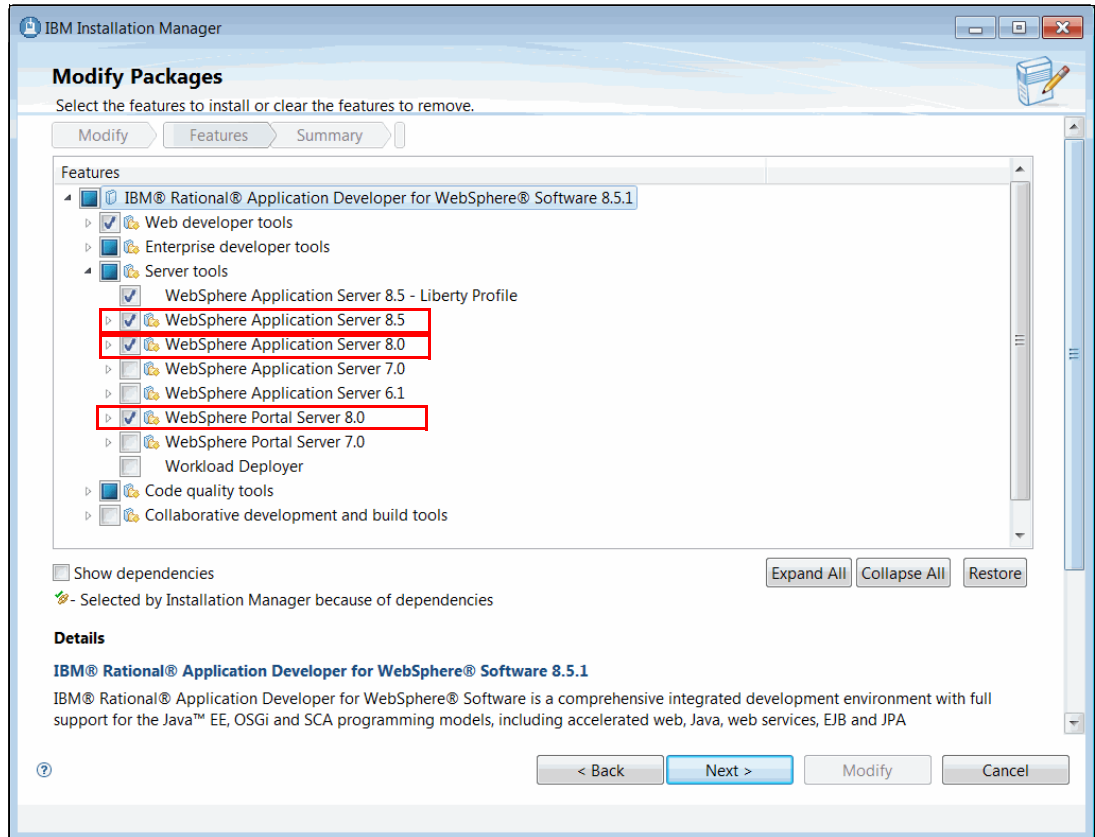


Figure B-3 Runtime components to be deployed with IBM Rational Application Developer 8.5.1

Importing the sample projects into IBM Rational Application Developer workspace

Each chapter in this book that includes samples has a correspondent subdirectory inside the archive SG248201.zip with the chapter's samples described in "Extracting the samples compressed file" on page 385.

Import the sample projects into the IBM Rational Application Developer workspace:

1. Locate the samples subdirectory for the chapter of interest. For example, for Chapter 9, "Integrating with external systems" on page 321, the corresponding subdirectory is \SAMPLES\8201Integration.
2. Open the IBM Rational Application Developer workspace and click **File** → **Import** → **Existing Projects into Workspace** → **Next** (see Figure B-4 on page 393).

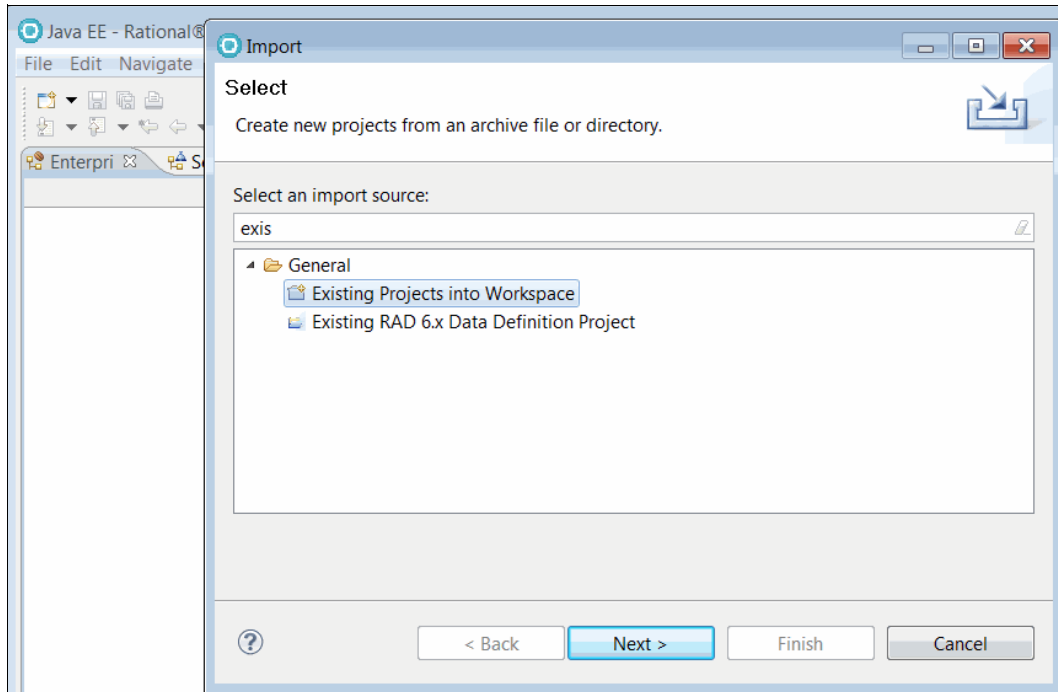


Figure B-4 Importing existing projects into the workspace

3. Click **Select root directory**, browse to the **SAMPLES** folder, and select the **8201Integration** folder. Click **OK** (see Figure B-5).

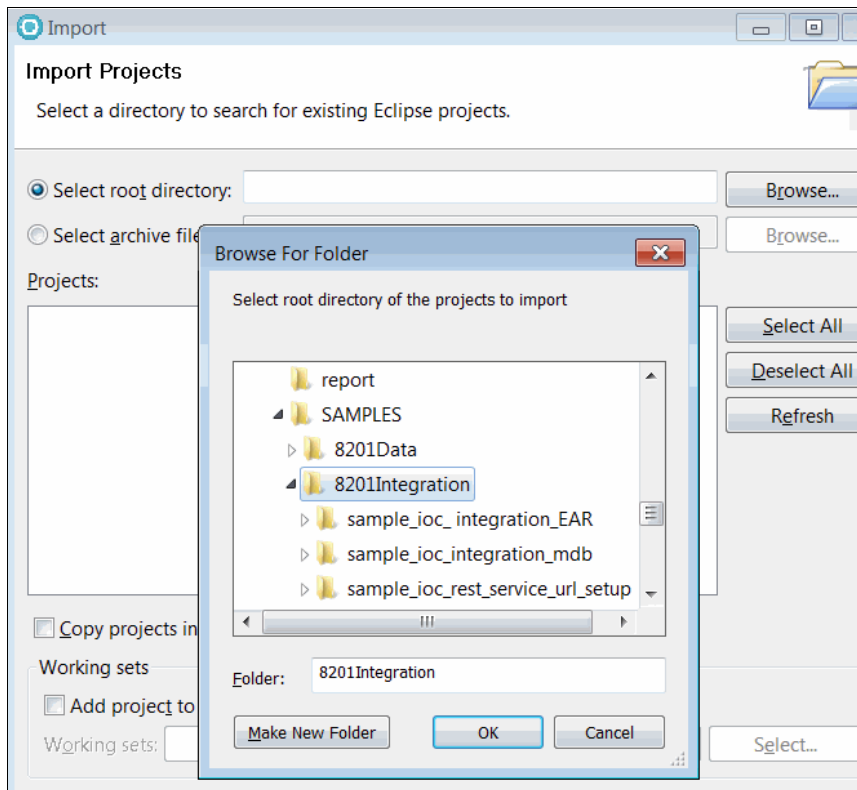


Figure B-5 Selecting 8201Integration folder for samples for Integrating with external systems chapter

4. Select all the projects as shown in Figure B-6 on page 394, and click **Finish**.

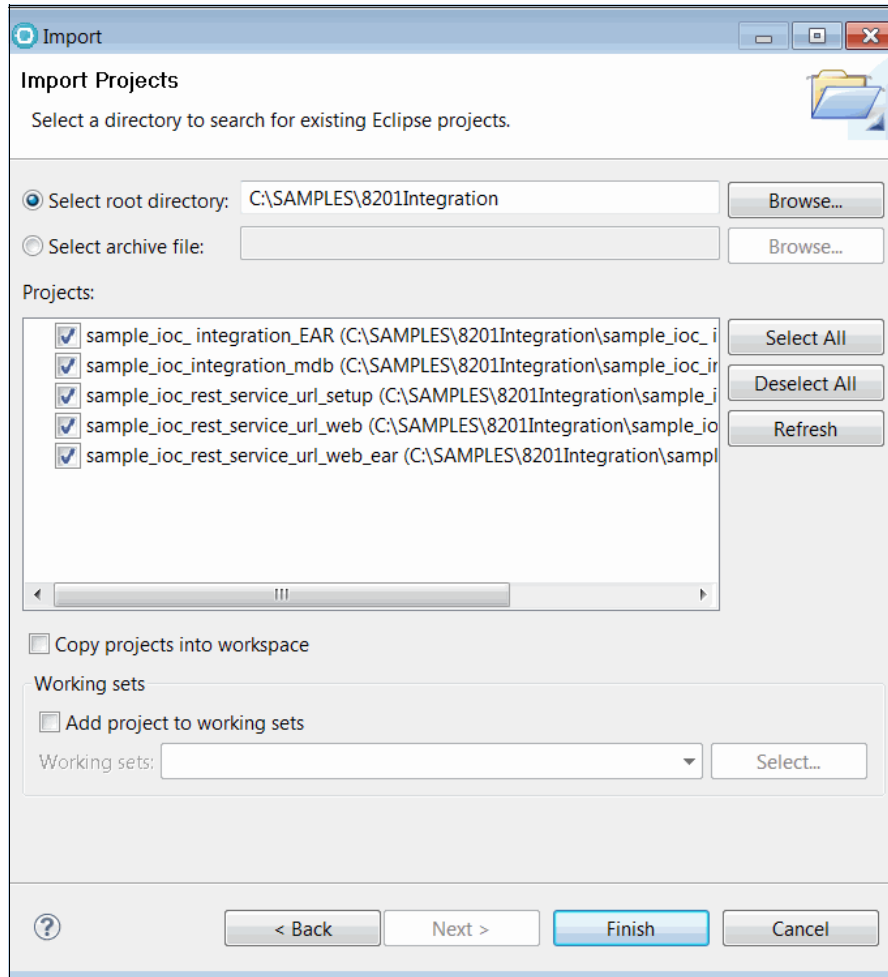


Figure B-6 Selecting all the samples for the Integrating with external systems chapter

All the samples for the chapter are now imported into the workspace, as shown in Figure B-7 on page 395.

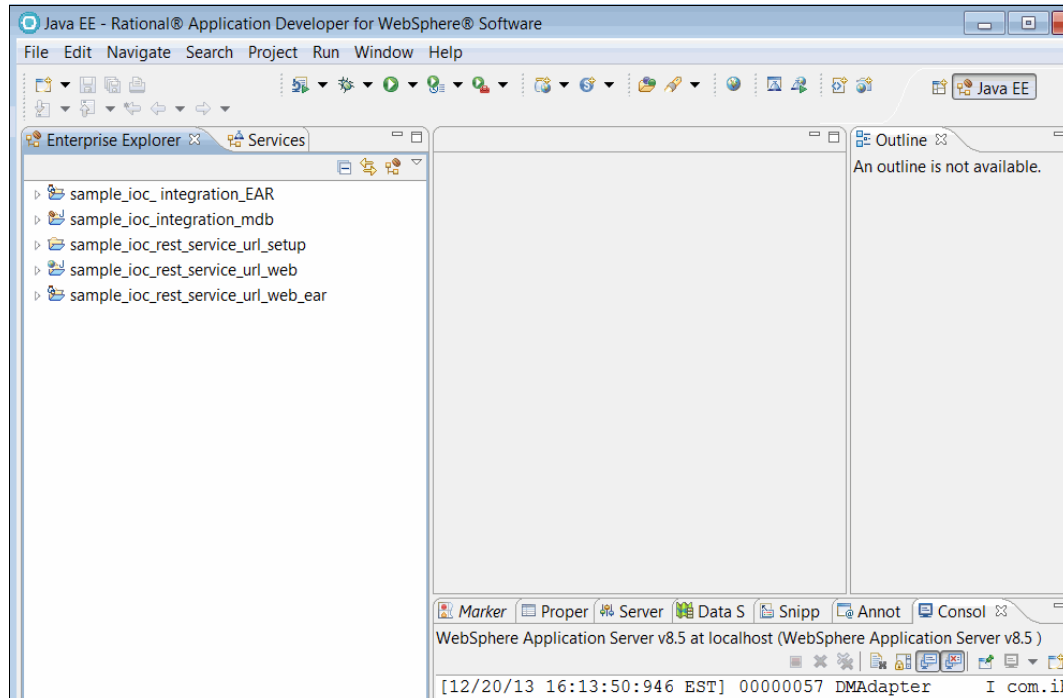


Figure B-7 Integrating with external systems sample projects imported into workspace

Installing IBM Intelligent Operations Center JARS in the development environment

The following samples require you to have specific IBM Intelligent Operations Center libraries configured in a build path:

- ▶ SAMPLES\8201PgmModel\:
 - sample_ioc_about_restore_portlet
 - sample_ioc_service_examples and sample_ioc_service_examples_ear
- ▶ SAMPLES\8201PortalVis\:
 - sample_ioc_contentviewer_web and sample_ioc_contentviewer_web_ear
 - sample_ioc_ui_custom_portlet
 - sample_ioc_ui_i18n_portlet and sample_ioc_ui_i18n_portlet_ear

Prepare the development environment for custom portlet development:

1. Download the IBM Intelligent Operations Center JAR files from the `lib` directory in the application server (by default, `/opt/IBM/ioc/lib`) to the development workstation.
2. Make the JAR files available in the development environment as a shared library:
 - a. Start IBM Rational Application Developer.
 - b. Click **Window** → **Preferences**.
 - c. From the left pane, expand **Java** → **Build Path**.
 - d. Click **User Libraries** (see Figure B-8 on page 396).
3. Click **New** to add a new user library, name it `IOC_JARS`, and click **OK**.
4. Select the `IOC_JARS` library and click **Add JARS**.

5. Browse to the JAR files that were downloaded from the IBM Intelligent Operations Center server in step 1 on page 395 and click **Open**. The window looks like Figure B-8. Click **OK**.

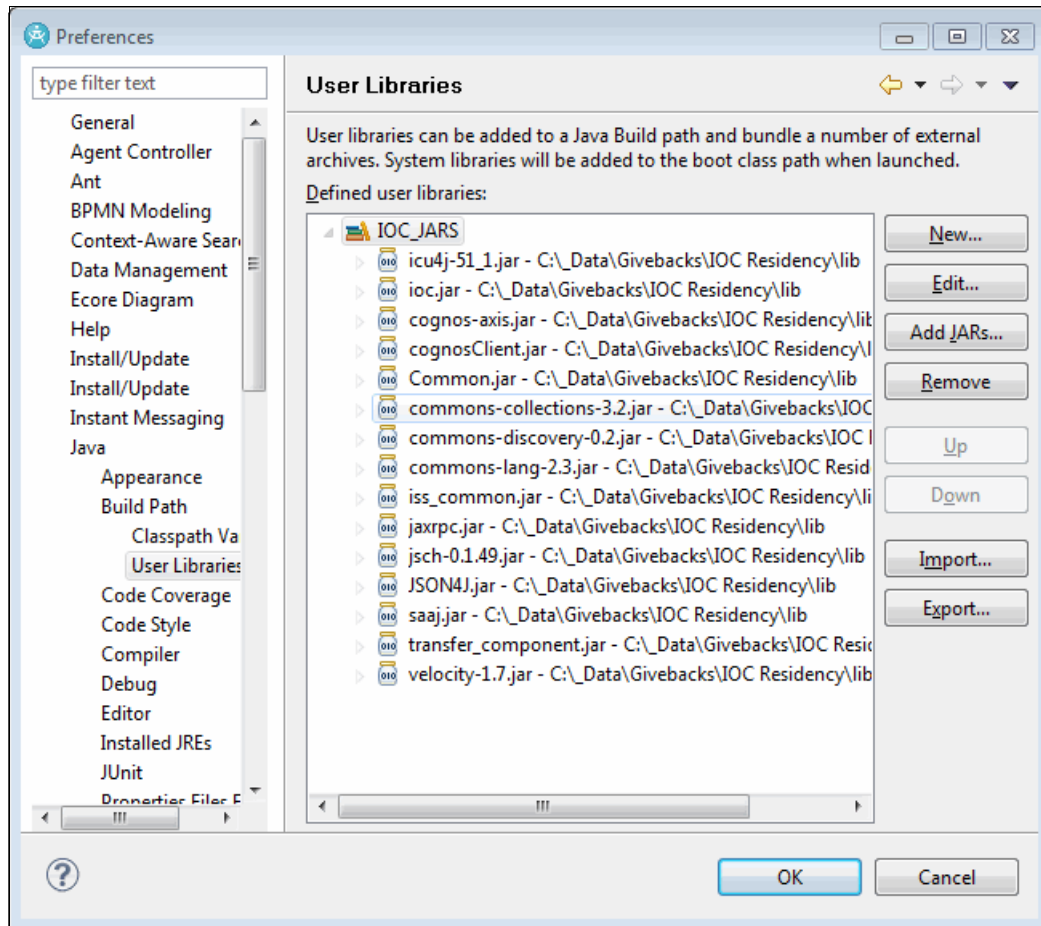
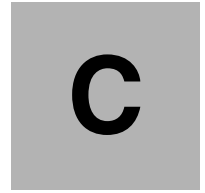


Figure B-8 IOC_JARS user library



Additional material

This book refers to additional material that can be downloaded from the Internet as described in the following sections.

Locating the web material

The web material associated with this book is available in softcopy on the Internet from the IBM Redbooks web server. Point your web browser at this website:

<ftp://www.redbooks.ibm.com/redbooks/SG248201>

Alternatively, you can go to the IBM Redbooks website at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG248201.

Using the web material

The additional web material that accompanies this book includes the following files:

<i>File name</i>	<i>Description</i>
SG248201.zip	Compressed code samples and sample data for data sources

System requirements for downloading the web material

The web material requires the following system configuration:

Hard disk space:	2 MB minimum
Operating system:	Windows/Linux

Downloading and extracting the web material

Create a subdirectory (folder) on your workstation, and extract the contents of the web material .zip file into this folder. For information about the sample directory and the included samples, see Appendix B, “Samples provided with this book” on page 383.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *IBM Intelligent Operations Center 1.5 to 1.6 Migration Guide*, SG24-8202
- ▶ *IBM Intelligent Operations Center for Smarter Cities Administration Guide*, SG24-8061
- ▶ *IBM Intelligent Operations Center for Smarter Cities*, REDP-4939

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ IBM Intelligent Operations Center 1.6 Information Center
<http://bit.ly/1fqLU09>
- ▶ IBM Intelligent Operations Center publication library
<http://www-01.ibm.com/support/docview.wss?uid=swg27039133>
- ▶ IBM Intelligent Operations Center V1.6 REST APIs
<http://www-01.ibm.com/support/docview.wss?uid=swg27039957>
- ▶ IBM Intelligent Operations Center Version 1.6.0.1 updates
<http://www-01.ibm.com/support/docview.wss?uid=swg24036406>
- ▶ Creating a DB2 data source
http://youtu.be/khnCVLq_wCE
- ▶ Creating a comma-separated value (CSV) data source
<http://youtu.be/4fnnr7w-hDE>
- ▶ Creating a Common Alerting Protocol (CAP) data source
<http://youtu.be/00mQ8phuSgo>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Redbooks

IBM Intelligent Operations Center 1.6 Programming Guide

(0.5" spine)
0.475" x 0.873"
250 x 459 pages



IBM Intelligent Operations Center 1.6 Programming Guide



Unleash the full potential of the programming model

Extend and customize the base solution

Learn from practical examples and sample code

IBM Intelligent Operations Center is an integrated solution. It provides a rich set of capabilities and line of business tools that business users with domain expertise and no technical background can use without customization.

IBM Intelligent Operations Center also provides services and extension points that developers can use to extend the IBM Intelligent Operations Center standard functions and develop capabilities specific to the domain and client requirements.

IBM Intelligent Operations Center includes an application-based programming model that supports all the interactions with the solution components. The programming model is based on industry standard Representational State Transfer (REST) and Java technologies. IBM Intelligent Operations Center includes a full set of REST and Java application programming interfaces (APIs) that provide a simplified development environment and make the platform easy to extend and customize for a large community of developers.

This IBM Redbooks publication gives a broad understanding of the IBM Intelligent Operations Center 1.6.0.1 programming model and available extension points. Many of the chapters describe working examples and usage scenarios that demonstrate how to extend the IBM Intelligent Operations Center base platform. This book includes sample code that can be downloaded from the IBM Redbooks website.

The target audience for this book consists of solution architects, developers, technical consultants, and solution administrators who will learn the following information:

- ▶ The options available to extend the IBM Intelligent Operations Center solution programmatically
- ▶ How to configure customizations tailored to specific client requirements
- ▶ How to use the available configuration tools to configure the solution without requiring programming

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**

SG24-8201-00

ISBN 0738439339