

webMethods Integration Cloud Help

Version 1.6.0

July 2015

This document applies to webMethods Integration Cloud Version 1.6.0 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2014-2015 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Table of Contents

New Registration	7
Creating an Account.....	8
Securing your Account.....	9
My Information	11
Editing Security Question and Answer.....	12
Managing Users	13
Adding Users.....	14
Updating User Information.....	16
Resetting Passwords.....	17
Managing Password Policies	19
Updating Password Policy Settings.....	20
Managing Access Profiles	23
Adding or Updating Access Profiles.....	24
Advanced Security	27
Add Keystore.....	28
Add Truststore.....	29
Add Partner Certificate.....	29
Company Information	31
Updating Company Information.....	32
Stages Management	35
Applying Access Profiles to a Stage.....	36
List of Applications	39
Adding Custom SOAP Applications.....	40
Managing Accounts	43
Adding or Editing Accounts.....	44
Account Configuration.....	46
Managing Operations	53
Adding or Editing Operations.....	54
Integrations	57
Creating Point-to-Point Integrations.....	58
Creating Orchestrated Integrations.....	60
Built-In Services.....	64
Date.....	64

Summary of Date elements.....	68
calculateDateDifference.....	69
dateBuild.....	70
dateTimeBuild.....	71
dateTimeFormat.....	72
getCurrentDateString.....	74
Documents.....	74
Summary of Document elements.....	75
findDocuments.....	75
Math.....	76
Summary of Math elements.....	77
absoluteValue.....	77
addFloatList.....	77
addFloats.....	78
addIntList.....	79
addInts.....	80
divideFloats.....	81
divideInts.....	82
max.....	82
multiplyFloatList.....	83
multiplyFloats.....	84
multiplyIntList.....	85
multiplyInts.....	86
randomDouble.....	86
roundNumber.....	87
subtractFloats.....	87
subtractInts.....	88
String.....	89
Summary of String elements.....	91
base64Decode.....	91
base64Encode.....	91
bytesToString.....	92
concat.....	93
indexOf.....	93
length.....	94
lookupDictionary.....	94
makeString.....	94
messageFormat.....	95
numericFormat.....	96
padLeft.....	97
padRight.....	98
replace.....	99
stringToBytes.....	100
substring.....	100
tokenize.....	101

toLowerCase.....	101
toUpperCase.....	102
trim.....	102
URLDecode.....	103
URLEncode.....	103
fuzzyMatch.....	104
Hashtable.....	105
Summary of Hashtable elements.....	105
containsKey.....	105
createHashtable.....	106
get.....	106
listKeys.....	106
put.....	107
remove.....	107
size.....	108
IO.....	108
Summary of IO elements.....	108
bytesToStream.....	109
streamToBytes.....	109
Reference Data.....	110
Pipeline and Signatures.....	111
Integration Details.....	111
Execution Results.....	114

1 New Registration

- Creating an Account 8
- Securing your Account 9

Registration is the process of creating a new Integration Cloud user account. You need to register to create your instance of the platform in the cloud.

Your organization may have multiple members, for example, your organization may be an entire company, an internal department, or just yourself. Similarly, your Integration Cloud account can have multiple internal users who interact with the platform. The very first person to open the Integration Cloud account becomes the first System Administrator for the tenant. The Administrator can then create new users (internal users).

Creating an Account

Creating an account is the first step in the Registration Process.

To create a new User Account

1. From the Integration Cloud login screen, click New Registration.
2. On the **Registration** page, complete the following fields:

Field	Description
First Name	Provide your first name. You can change the value after the user is created from the Settings > Users screen.
Last Name	Provide your last name. You can change the value after the user is created from the Settings > Users screen.
Company	Provide your company name. You can change that later from the Settings > Company Information screen.
Country	Provide your country name. You can change that later from the Settings > Users screen.
State or Province	Provide your State or Province name. You can change that later from the Settings > Users screen.
Phone	Provide your phone number. You can change that later from the Settings > Users screen.

Field	Description
Sub-Domain for Service Portal	Provide a unique sub-domain, typically your company name. For example, suppose you are at ABC Company and you decide to use “abc” as your unique sub-domain. With that setting, you will access your instance of the platform at https://abc.webmethodscloud.com.
Email Address	Provide your email address. The email field becomes both the user name and the email address for the initial user. You can change the values after the user is created, from the Settings > Users screen.
Promo Code	Enter a valid promotion code if you have one, for availing subscription benefits.
I agree to the Terms of Service	Select this option to agree to the webMethods Integration Cloud Terms of Service.

3. Click Submit to continue to the next step to activate and secure your account or click Cancel to go back to the Login screen. After you click Submit and as soon as the registration process is complete, an email will be sent to the email address you provided during registration. Use the temporary password sent in the email to log in. You will be asked to change your password.

Securing your Account

Securing your account is the second step in the Registration process. When you Login for the first time, you are asked to select a security question. This security question and answer is associated with your user name. If you forget your password, this information is used to verify the account ownership.

To secure your account

1. Select a security question from the drop-down list. Optionally, you can select the option “Write my own security question” to compose a personalized security question.
2. Provide an answer to the security question.
3. Click **Submit**.

Note: If you forget your password, in the Login page, click the **Forgot Password?** link, enter your user name, and then click **Submit**. An email is sent that contains a request to answer the Security Question you chose when your account was created. When the email arrives, click the link to open the

Password Reset page. Provide the answer to your Security Question and enter a new password. After you provide the correct answer, you can log in with your changed password.

2 My Information

■ Editing Security Question and Answer	12
--	----

This screen allows you to view the profile information for the logged in user for the Integration Cloud instance.

You can view the **Basic Information**, **Locale Information**, and the **Address and Contact Information**.

Click the **Edit** button to update the information.

Editing Security Question and Answer

To update the Security Question and Answer

1. From the Integration Cloud navigation bar, go to <User name> > My Profile > Security Question.
2. Select a Security Question and type a Security Answer. You can change the Security Question associated with your Account Login/Password.
3. Click **Submit**.

Note: The User name and Email address can differ, depending on the settings specified in the **My Information** screen.

3 Managing Users

- Adding Users 14
- Updating User Information 16
- Resetting Passwords 17

You can use the **Users** screen to create and manage administrators and other users. A User has a login identity, password, email address, and other descriptive attributes.

From the main **Users** screen, you can search for users, create a new user, update existing user information, and reset a user's password.

Users who have the **User and Ownership Controls** permission under **Settings > Access Profiles > Administrative Permissions > User and Ownership Controls** can edit the user information.

Adding Users

You can add users for accessing the platform. The operations that a user can perform is determined by their *Access Profile*.

To add a user

1. From the Integration Cloud navigation bar, go to Settings > Users.
2. From the upper right part of the Users screen, click Add New User.
3. On the **Basic** tab, complete the following fields. Required fields are marked with an asterisk in the screen.

Field	Description
First name	User's first name as it should appear in the platform.
Last name	User's last name as it should appear in the platform.
Title	User's professional title.
Access Profile	<p>The access profile assigned to the User. Each User is assigned an access profile, which can be shared by other users. An Access Profile specifies the network locations (IP addresses) from where it is possible to login and administrative permissions. Specify one of the following Access Profiles:</p> <ul style="list-style-type: none"> ■ Administrator - Provides permissions needed by the System Administrator. ■ Regular User - Provides permissions that are more appropriate for normal users.

By default, the system administrator can change the Administrative Permissions associated with each Access Profile (except the above mentioned **Administrator** Access Profile), and can add additional Access Profiles, as needed.

Field	Description
	<p>Note: By default, the Administrator and Regular User Access Profiles are associated with the Development Stage. If you have created a new Access Profile, ensure that the Access Profile you have created is associated with the Development Stage. See Adding or Updating Access Profiles for more information.</p>
Employee Number	Optional identification number for each employee.
Email	Email address of the user. The email address is used to send the welcome and login information to the user.
User name	User name is a unique name associated with each User and is required to log in. It can be an email address or an alphanumeric text string.
Active	Select this option to indicate that the user account is active. You can use this option to reactivate a locked or disabled user account.
Force password change on first login	Select this option to force the user to change the login password when logging in for the first time. This option is available only when creating a new user.

4. On the **Locale** tab, complete the following fields:

Field	Description
Time Zone	Choose a Time Zone Code from the drop-down list.
Date Format	Choose a Date Format from the drop-down list. "mm" is "Month", "dd" is "Day", and "yyyy" is Year. Dates and Times are used throughout the platform, in Appointments, as Start/End Dates in Tasks, Expected Close Date, Estimated Start/End Date, Date Due, and so on. Default formats are specified under the Settings > Company Information > Advanced Information tab. Administrators and Users can change the default selection in the Users screen.
Time Format	Select a 12-hour clock (hh:mm a) with AM/PM, or a 24-hour clock (HH:mm).

Field	Description
Locale	Select the user's locale setting. This setting determines the format for numbers, decimal fields, and percentages.

5. On the **Address and Contact** tab, complete the following fields:

Field	Description
Phone	Primary phone number for the user.
Mobile Phone	Mobile phone number for the user.
Fax	Fax number for the user.
Street Address	Street address for the user.
City	City for the user.
State/Province	State or province for the user.
Postal/Zip Code	Postal or ZIP Code for the user.
Country	Country for the user.

6. Click **Add** if you are adding a User or **Apply** if you are editing any User information.

You can fill the **Address and Contact** section later or the Administrator can fill the details by editing the record after the User has been added. The **Address and Contact** screen is also available under **<User name> > My Profile > My Information** tab.

Note: A User can log in and then go to **My Profile** to change the user details. The Administrator who created the User can also edit the User details.

Updating User Information

To edit or update the user information

1. From the Integration Cloud navigation bar, click Settings > Users.
2. Select a user from the list and then click Edit.
3. Make necessary modifications. See "[Adding Users](#)" on page 14 for information on the relevant fields. You can also enter or update the following information on the **Address and Contact** tab. Required fields are marked with an asterisk in the screen.

Field	Description
Phone	Primary phone number for the user.
Mobile Phone	Mobile phone number for the user.
Fax	Fax number for the user.
Street Address	Street address for the user.
City	City for the user.
State/Province	State or province for the user.
Postal/Zip Code	Postal or ZIP Code for the user.
Country	Country for the user.

4. Click **Apply**.

The default initial information comes from the *Company Information* page, but you can modify it here.

Note: A user can log in and then go to **My Profile** to change the user details. The administrator who created the user can also edit the user details.

Resetting Passwords

To reset a User password

1. From the Integration Cloud navigation bar, go to Settings > Users.
2. For the User whose password is to be reset, select the user and click **Reset Password**.

Integration Cloud sends an email notification to the user regarding the password reset.

Note: A User can log in and then go to **My Profile** to change the user details. The administrator who created the User can also edit the User details.

4 Managing Password Policies

■ Updating Password Policy Settings	20
---	----

A Password Policy defines password requirements and login protections. Users who have the **Access Control** permission under **Settings > Access Profiles > Administrative Permissions > User and Ownership Controls** can edit the Password Policy information.

You can view the password policies for the Integration Cloud instance in this screen.

Click **Edit** to modify Password Policy information.

Updating Password Policy Settings

You can set password policies for users on the **Update Password Policy** screen.

To update the Password Policy

1. From the Integration Cloud navigation bar, click Settings > Password Policy.
2. Click Edit.
3. On the **Update Password Policy** screen, make the necessary modifications.

Field	Description
Minimum Length	Specify the minimum number of characters in the password. Default: 6 characters. Range: 6-10 characters.
Required Character types	This option defines the level of security for passwords, which can be simple and allow any character combination, or very secure, requiring upper and lower case characters, as well as special characters. Default: No Restrictions.
Expires in	Specify the number of days the password will remain valid before the user will be prompted to change it. Default: 90 Days. Range: 15, 30, 60, 90, 120 days, Never. By default, no user is exempt from the Password Policy. You can specify a user to be excluded from the password expiration policy by selecting <i>Never</i> .
New Password cannot match	The new password cannot match the number of previous passwords. Default: Last Password. Range: Last 2-5 passwords.
Minimum Age	Specify the number of days that must pass before a user can change passwords. Default: No Minimum. Range: 1-5 Days.
Account Lockout Threshold	Specify the number of login attempts before the account is locked out. Default: 5 failed tries. Choices: 3-10 failed tries, No limit.

Field	Description
	<p>The login limit defines the number of failed attempts allowed before a user account is disabled or locked for a specified time. When a user attempts to login and fails (because of an incorrect password), each attempt counts against the login limit. When the login limit is achieved, the account is disabled or locked for a specified time, according to the parameters set in the <i>Account Lockout Duration</i> field. The login limit is defined by the <i>Password Policy</i>.</p>
Account Lockout Duration	<p>Specify the length of time that an account is locked out. Default: 15 minutes. Choices: 5, 10, 15, 30 minutes, 1 hour, Disable.</p>
Record Information	<p>For audit purposes, the following information is displayed after you save the record:</p> <p>Last Modified By <username> on {date} <time> Created by System.</p>

4. Click Apply.

5 Managing Access Profiles

- Adding or Updating Access Profiles 24

An *Access Profile* specifies a collection of permissions that can be applied to multiple users. Each user is assigned an Access Profile, which can be shared by other users.

Users who have the **User and Ownership Controls** permission under **Settings > Access Profiles > Administrative Permissions > User and Ownership Controls** can edit the Access Profiles information.

An Access Profile specifies:

- The network locations (IP addresses) from where it is possible to login.
- Administrative permissions. If checked, the user is granted all administrative permissions and if left unchecked, a list of administrative permissions is displayed. Permissions can then be individually selected.

The default Access Profiles are:

- Administrator, which provides permissions needed by the System Administrator.
- Regular User, which provides permissions that are more appropriate for normal users.

By default, the system administrator can change the Administrative Permissions associated with each Access Profile, and can add additional Access Profiles, as needed.

To edit an existing Access Profile, select the profile and click **Edit**. To delete an Access Profile, select the profile and click **Delete**. To create a new Access Profile, click **Add New Access Profile**.

Adding or Updating Access Profiles

Use the **Access Profiles** screen to create or edit profiles assigned to users.

To add or update an Access Profile

1. From the Integration Cloud navigation bar, go to Settings > Access Profiles.
2. Click Add New Access Profile to add a custom access profile or click Edit to change any field in an existing Access Profile.
3. On the **Add New Access Profile** or **Update Access Profile > Access Profile Information** tab, complete the following fields. Required fields are marked with an asterisk in the screen.

Field	Description
Name	Provide a name for the Access Profile. You can reference the profile by name when assigning it to a user.
Description	Provide a general description for the Access Profile.

4. On the **Login IP Access Restrictions** tab, complete the following fields:

Field	Description
IP Address Ranges	<p>For extra security, enter ranges of IP addresses from which users are allowed to access the platform. If a user attempts to login from a computer on a network outside of the specified range, access to the platform is denied.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note: A maximum of 25 IP address ranges can be specified. Accepted format is xxx.xxx.xxx.xxx - yyy.yyy.yyy.yyy, where xxx and yyy are numbers in the range 0-255 and xxx.xxx.xxx.xxx is less than or equal to yyy.yyy.yyy.yyy. To specify a single IP address, use the same IP address for the start and endpoint of the range: 192.168.1.1 - 192.168.1.1</p> <p>When a user attempts to log in, the IP address of the system the request originated from is checked against the configured settings. If the address is in the allowed range, the user can continue the login process. Otherwise, login is denied. Access violations are recorded in the audit log, identifying both the user and the IP address from where the login attempt originated. Login restrictions do not apply to Customer Support logins.</p> </div>

5. On the **Administrative Permissions** tab, select the operations a user can perform in order to view, create, update, and delete permissions and to allow the user to customize selected aspects of the platform.

Field	Description
User and Ownership Controls	<p>User Management - Select this option if you want to add, update, delete users, or assign users to Access Profiles.</p> <p>Access Control - Select this option if you want to allow a user to modify Access Profiles, specify user application access rights, manage Access Profiles, or specify the password policy.</p> <p>Manage Personal Setup - Select this option if you want to allow a user to modify the personal information.</p>
Account Controls	<p>Manage Company Capabilities - Select this option if you want to allow users to modify the company information.</p>
Functional Controls	<p>Select the required options under Accounts, Operations, Integrations, Stages, Advanced Security, and Reference Data. You must select the</p>

Field	Description
	required permissions to create, update, delete, or administer those functions.

6 Advanced Security

■ Add Keystore	28
■ Add Truststore	29
■ Add Partner Certificate	29

Keystores and truststores are files that function as repositories for storage of keys and certificates necessary for SSL authentication, encryption/decryption, and digital signing/verification services. Keystores and truststores provide added layers of security and ease of administration, compared to maintaining the keys and certificates in separate files.

Integration Cloud stores its private keys and SSL certificates in keystore files and the trusted roots for the certificates in truststore files. Keystores and truststores are secure files with industry-standard file formats.

If you want to run services that submit HTTPS requests to other resources on the Internet, your server will be acting as a client and will receive certificates from these resources. In order for these transactions to work, your server must have copies of their public keys and signing CA certificates.

To identify a particular keystore or truststore file, or private key within a keystore, aliases are used. The use of aliases simplifies keystore and truststore management, because you do not need to enter path information when specifying a keystore, truststore, or the private key.

Note: You can add, edit, or view keystore and truststore aliases and partner's self-signed certificates in the **Settings > Advanced Security** tab and can use them to secure your Custom SOAP Application Account. Users who have the **Administer** permission under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Advanced Security** can add, edit, and delete Keystores, Truststores, and Partner Certificates.

To add a Keystore, from the Integration Cloud navigation bar, click **Settings > Advanced Security > Keystores > Add Keystore**.

To add a Truststore, from the Integration Cloud navigation bar, click **Settings > Advanced Security > Truststores > Add Truststore**.

To add a Partner Certificate, from the Integration Cloud navigation bar, click **Settings > Advanced Security > Partner Certificates > Add Certificate**.

Add Keystore

Integration Cloud allows you to upload a Keystore file to store SSL certificates and keys. A Keystore file contains one or more pairs of a private key and signed certificate for its corresponding public key. From this screen, you can create aliases for the Keystore, so that they can be referenced while creating an Account for a Custom SOAP Application.

To add a Keystore

1. From the Integration Cloud navigation bar, click **Settings > Advanced Security > Keystores > Add Keystore**.
2. Provide a name and description for the Keystore file.

3. In the “Type” field, select the Keystore file format. The default file format is JKS. You can also use PKCS12, a commonly used, standardized, certificate file format that provides a high degree of portability.
4. Click “Browse” to select the Keystore file.
5. In the “Passphrase” field, enter the passphrase for the Keystore file. The passphrase must have been defined at the time the Keystore was created.
6. Click “Next” to protect the Key Aliases with passphrases. A key alias is a label for specific key within a Keystore. Enter a passphrase for each Key Alias found in the Keystore file and then click “Finish” to upload the Keystore file.

The uploaded Keystore file can be used while creating an Account for a Custom SOAP Application.

Add Truststore

Integration Cloud allows you to upload a Truststore file, which contains the trusted root of the certificate or signing authority (CA). From this screen, you can create aliases for the Truststore, so that they can be referenced while creating an Account for a Custom SOAP Application.

To add a Truststore

1. From the Integration Cloud navigation bar, click **Settings > Advanced Security > Truststores > Add Truststore**.
2. Provide a name and description for the Truststore file.
3. In the “Type” field, select the Truststore file format. The default file format is JKS. You can also use PKCS12, a commonly used, standardized, certificate file format that provides a high degree of portability.
4. Click “Browse” to select the Truststore file.
5. In the “Passphrase” field, enter the passphrase for the Truststore file. The passphrase must have been defined at the time the Truststore was created and is used to protect the contents of the Truststore.
6. Click “Save” to upload the Truststore file.

The uploaded Truststore file can be used while creating an Account for a Custom SOAP Application.

Add Partner Certificate

Integration Cloud allows you to upload the Partner’s certificate which contains its public key. The Partner’s certificate with the public key is required to encrypt outbound request messages and verify the signature of inbound messages.

From this screen, you can create aliases for Partner Certificates, so that they can be referenced while creating an Account for a Custom SOAP Application.

To add a Partner Certificate

1. From the Integration Cloud navigation bar, click **Settings > Advanced Security > Partner Certificates > Add Certificate**.
2. Provide a name and description for the Partner Certificate file.
3. Click "Browse" to select the Partner Certificate file.
4. Click "Save" to upload the Partner Certificate file.

The uploaded Partner Certificate can be used while creating an Account for a Custom SOAP Application.

7 Company Information

■ Updating Company Information	32
--------------------------------------	----

This screen specifies your company information. Users who have the **Manage Company Capabilities** permission under **Settings > Access Profiles > Administrative Permissions > Account Controls** can edit the company information.

Click **Edit** to update the company information.

Updating Company Information

You can view and update the company information and use them across all applications in the platform.

To update the Company Information

1. From the Integration Cloud navigation bar, go to Settings > Company Information.
2. Click Edit.
3. On the **Basic** tab, complete the following fields. Required fields are marked with an asterisk in the screen.

Field	Description
Tenant ID	This is the unique ID assigned to your organization's tenancy on the platform. This field cannot be edited and appears in view only mode under the Basic tab.
Company Name	The name of the company. This field accepts only alphanumerics, spaces, and hyphens (-). The company name is automatically populated from the Registration screen.
Street	The street address of the company.
City	The city where the company is located.
State/Province	The state or Province where the company is located. The state or province name is automatically populated from the Registration screen.
Postal/Zip Code	The postal or zip code for the company.
Country	The country where the company is located. The country name is automatically populated from the Registration screen.
System Notification	Enter an address or comma-separated email addresses to receive system notifications. Such notifications can occur, for example,

Field	Description
Email Addresses	when a connection to the system mailbox fails after repeated attempts. This field displays the information from the Registration screen but you can change that later using the Edit button.

4. On the **Advanced Information** tab, complete the following fields:

Field	Description
Time Zone	Choose your time zone from the drop-down list.
Time Format	Choose a time format from the drop-down list. You can choose a 12-hour clock with AM/PM or a 24-hour clock. hh:mm a - 12-hour clock - 3:30 AM, 3:30 PM HH:mm - 24-hour clock - 3:30, 15:30
Date Format	Choose a date format from the drop-down list. mm is "Month", dd is "Day", yyyy is Year and the delimiters are:(/) slash or stroke(-) dash or hyphen(.) period, dot, or full stop.
Default Locale	The field displays the user's initial locale setting and determines the format for numbers, decimal fields, and percentages. Choose any other locale from the drop-down list.
Log All Record Access	Select this field to log all record access attempts in the Audit Log. The Audit Log tracks all changes to data (additions, modifications, or deletions) and customizations to the platform.
Log All Access Violations	Select this field to log all access violations in the Audit Log. The Audit Log displays a full history of actions in the platform. Most access violations occur when a user attempts to access an element in the platform, but does not have the appropriate permissions.
Last Modified	This field displays the date and time when the company information record was last updated and cannot be edited.

8 Stages Management

- Applying Access Profiles to a Stage 36

Stages provide safe environments for development and testing that are separated from the production environment. They allow Integrations to migrate from the development environment to the production environment through the intermediate environments.

Integration Cloud provides ways to manage the life cycle of an Integration service development. The typical life cycle of an Integration service development involves creating Integrations, testing them, and making them production worthy. Each of these activities can be termed as different stages of an Integration service life cycle development. To aide these activities, Integration Cloud provides you with *Stages*.

Your organization is a tenant in the platform. When you log in to the platform, you log into your organization's tenancy. When you set up a stage, an environment is created for testing and executing the Integrations in the production environment. You can also run the scheduler to schedule Integrations.

A predefined set of stages is allowed, each representing an activity in the Integration service life cycle development. They are:

- Development
- Test
- Pre-Live
- Live

You can create or delete stages only in a particular order. By default, every user gets a *Development Stage*. In the Development Stage, you can create, update, delete, or view Integrations. In other stages, Integrations can be pulled from a preceding stage or deleted. Further, Integrations can be pulled into a stage only from a preceding stage.

Note: Users who have the Administer permission under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Stages** can add or delete the Stages information.

To add a new stage, click **Add New Stage**.

Applying Access Profiles to a Stage

The typical life cycle of an Integration service development involves creating Integrations, testing them, and making them production worthy. Each of these activities can be termed as different stages of an Integration development.

Every stage can be assigned a number of Access Profiles and users who are assigned the required Access Profiles can perform activities on that stage. For example, if in an Access Profile, **Execute Integrations** permission is granted, then the user assigned with that Access Profile can execute Integrations on the stages to which the Access Profile is assigned. If the Access Profile needs to perform scheduling activity on the Live stage, then the Access Profile needs to have access to that stage as well. The Development stage can be accessed by everyone.

Click **Add** to add the next stage. Multiple boundary arrows indicate that more stages can be added.

Note: The Accounts drop down list in the Live stage lists all the Accounts defined in the Development stage. Accounts that are not present in the Live stage are highlighted. Click on such an Account in the Live stage to view the **Edit Accounts** page. Only active or enabled Accounts are listed in the drop down list.

Click **Delete** to delete a stage. You cannot delete the **Development** stage.

Note: When a stage is deleted, everything it contains is erased and cannot be recovered.

To apply Access Profiles to a stage

1. From the Integration Cloud navigation bar, click Settings > Stages Management. All stages added including the Development stage is displayed. Initially, before any other stages are added, the Development stage is displayed.
2. Click the Access Profiles icon and select the Access Profiles you want to apply to the desired stage.

Note: By default, the **Administrator** and **Regular User** Access Profiles are associated with the Development Stage. If you have created a new Access Profile, ensure that the Access Profile you have created is associated with the Development Stage.

3. Click **Apply**.

The Access Profiles are applied to the selected stage.

Note: You can pull Integrations from all other stages except from the **Development** stage. An Integration service depends on the Action, Trigger, and Account. When an Integration is pulled, all its dependents will also be pulled and copied to that stage. If you update only the action or trigger service, then only those will be pulled into the next stage.

9 List of Applications

■ Adding Custom SOAP Applications	40
---	----

Integration Cloud allows you to create and govern Integrations between Software as a Service (SaaS) or on-premise applications. A set of predefined and configurable Applications are provided, for example, Salesforce, StrikeIron, ServiceNow, and so on, which enable you to connect to the particular SaaS providers.

Custom SOAP Applications, which are used to access third party Web Services are also listed in this page. To create a Custom SOAP Application, click **Add New Application**.

On-Premise applications loaded from on-premise systems are also listed in the **List of Applications** page, but you will not be able to create Accounts or Operations for on-premise applications. Those can be uploaded only from the webMethods Integration Server. Further, when you upload services as part of an application from the on-premise Integration Server to webMethods Integration Cloud, the comments field of the service is uploaded and displayed in the webMethods Integration Cloud application. This field will be displayed if present and cannot be edited. See the *Configuring On-Premise Integration Servers for webMethods Cloud* document for more information.

You can also create Accounts and Operations for an Application and Integrations between different SaaS applications from the **List of Applications** page. For any Application, you can select **Accounts**, **Operations**, or **Integrations** if you want to create or edit them for that Application.

To use an Application, you are required to agree to the summary of terms. Click **I agree** to use the Application. Click **I do not agree** if you disagree with the summary of terms and do not want to use the Application. Click **Cancel** to go back to the **List of Applications** page.

Note: Users who have the **Functional Controls** permission under **Settings > Access Profiles > Administrative Permissions > Functional Controls** can create, update, administer, execute, deploy, or delete the Accounts, Operations, Integrations, Stages, Advanced Security, and Reference Data information.

Adding Custom SOAP Applications

The “List of Applications” screen allows you to create a Custom SOAP Application. The Custom SOAP Application enables you to access third party Web Services hosted in the cloud or on-premise environment. The Custom SOAP Application uses a WSDL to create consumer operations and can be a trigger or action or both.

The following features are supported for Custom SOAP Applications:

- A Web Service implementation that follows the WS-I Basic Profile 1.1 specification.
- Custom SOAP Applications can be created with a WSDL that is accessible through publicly or locally accessible URLs.
- Web Services that use WS-Security. Custom SOAP Applications can be created with WSDLs that are annotated with WS-Security Policy/Policies.

- Web Services with SOAP version 1.1 and 1.2 and Style/Use as Document/Literal and RPC/Literal (RPC/Encoded model is not supported for SOAP version 1.2).
- The following SOAP Binding types are supported:
 - SOAP over HTTP.
 - SOAP over HTTPS.
- Authentication type: HTTP Basic Token.

Custom SOAP Applications have the following restrictions:

- The WSDL and associated schema(s) must be accessible through a publicly or locally accessible URL.
- Only WSDLs with WS-Security policies are supported. Any other policies, for example, WS-Addressing, WS-Reliable Messaging, and so on, are not supported. If you create Custom SOAP Applications with WSDLs having non-WS-Security Policies, exceptions may appear while executing Integrations.
- Manual addition of WS-Security Policies in a Custom SOAP Application is not supported. Custom SOAP Applications with WS-Security can be created with only policy-annotated WSDLs, that is, WSDLs that already have WS-Security Policies annotated in them.
- Custom SOAP Applications cannot be created by uploading a WSDL file. You can create a Custom SOAP Application only by using a valid WSDL URL, which can be accessed over the network.
- SOAP over JMS is not supported.
- Only Basic Authentication is supported. Other authentication types such as Digest, NTLM, and Kerberos are not supported.
- You will not be able to attach or upload a file while executing an Integration.

To add a Custom SOAP Application

1. From the Integration Cloud navigation bar, click “Applications”.
2. Click “Add New Application”.
3. Provide a name and description of your Custom SOAP Application. The description you enter here will appear in the “List of Applications” page. Required fields are marked with an asterisk in the screen.
4. Select the “Type”. For example, select SOAP if you want to create a SOAP-based Application.
5. Click Next and enter the URL for a supported WSDL document in the “WSDL URL” field. The URL should begin with http:// or https://. The URL is used to retrieve the WSDL for the Web Service.
6. Enter the user name and password in the “Authentication” section. This is used to verify the consumer's authentication credentials contained in the request's

Authorization header. The user/password value in the Authorization header must be authenticated as a valid user. The credentials are authorized against the list of registered users.

7. Click "Next" to review the details you have entered and then click "Finish" to create your Custom SOAP Application.

10 Managing Accounts

- Adding or Editing Accounts 44

This screen lists all the available Accounts. If you have configured the Account details incorrectly in any stage, the stage will appear in red color in the **Connectivity Status** column. If an Account is configured correctly in a particular stage, then the stage will appear in green color and if an Account is not configured in a particular stage, then the stage will appear in white color.

For on-premise Applications, select an Account and then click **Test Connection** to show the status of the connection. The Account can be used to execute services on the on-premise Integration Server. See the *Configuring On-Premise Integration Servers for webMethods Cloud* document for information on how to configure Integration Server as an on-premise server for use with Integration Cloud.

Note: Only enabled or active Accounts are listed in the drop down list of the Operation wizard, Integration wizard, Look up Transformer, and Stages Management.

You can create, edit, or delete an Account for a particular application.

Note: Users who have the required permissions under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Accounts** can create, update, or delete the Accounts information.

To create or edit an Account

1. From the Integration Cloud navigation bar, click “Applications”.
2. Select an application from the list and then click **Accounts**.

To use an Application, you are required to agree to the summary of terms. Click **I agree** to use the Application. Click **I do not agree** if you disagree with the summary of terms and do not want to use the Application. Click **Cancel** to go back to the **List of Applications** page.

3. From the Accounts screen, click Add to add an Account or click Edit to update an existing Account.

Adding or Editing Accounts

Use the **Accounts** screen to add or update Accounts. The options available may vary according to the selected Application.

To add or edit an Account

1. From the Integration Cloud navigation bar, click Applications.
2. Select an Application from the list and then click Accounts.

To use an Application, you are required to agree to the summary of terms. Click **I agree** to use the Application. Click **I do not agree** if you disagree with the summary of

terms and do not want to use the Application. Click **Cancel** to go back to the **List of Applications** page.

3. From the Accounts screen, click "Add New Account" to add an Account or click Edit to change any field in an existing Account.
4. On the New Account or Edit Account screen, complete the following fields. Based on the Application you had selected, applicable fields are displayed. Required fields are marked with an asterisk in the screen.

Field	Description
Save As	Provide a valid name for the Account. This field is common for all Applications.
Description	Provide a description for the Account. This field is common for all Applications.

The Account configuration section allows you to provide details to connect with the Application. The fields available may vary according to the selected Application. If you have added any stage in the "Stages Management" page, the stages will appear as tabs in the Account configuration section. Enter the Account configuration details for each stage. If you have configured the Account details incorrectly in any stage, the stage will appear in red text and the Account will be inactive. If an Account is configured correctly in a particular stage, then the stage appears in green text and is active. Only active or enabled Accounts are listed in the drop down list of the Operation wizard, Integration wizard, Look up Transformer, and Stages Management.

See "[Stages Management](#)" on page 35 for more information.

Note: You must have the permission to administer stages (**Access Profiles > Administrative Permissions > Functional Controls > Stages**) if you want to create or delete stages.

Click on the following links for information on the Account configuration fields:

List of Applications

- ["Custom SOAP Application" on page 46](#)
- ["Amazon Simple Storage Service \(S3\)" on page 48](#)
- ["Amazon Simple Queue Service \(SQS\)" on page 49](#)
- ["Salesforce CRM" on page 49](#)
- ["Salesforce Bulk Data Loader" on page 50](#)
- ["SuccessFactors HCM" on page 50](#)
- ["ServiceNow Enterprise Service Management" on page 51](#)

Field	Description
-------	-------------

- ["StrikeIron Contact Verification" on page 51](#)

- Click Save or Update to save your settings or click "Save All Stages" to save the changes done in all the stages.

A new Account will be created.

Account Configuration

Custom SOAP Application

Field	Description
-------	-------------

Protocol The protocol to use for the port (HTTP or HTTPS). If you select HTTPS, the **Keystore Alias** and **Key Alias** fields appear.

Host Host name or IP address of the server on which the Web Service resides.

Port An active HTTP or HTTPS type listener port defined on the host server specified in the **Host** field.

Port Binding The specific bind address to which to bind this port.

User User name used to authenticate the consumer at the HTTP or HTTPS transport level on the host server.

Password The password used to authenticate the consumer on the host server.

Keystore Alias Alias to the keystore that contains the private key used to connect to the Web Service host securely. This field applies to the HTTPS transport type only. You can also add a new Keystore from this field.

Note: Users who have the Administer permission under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Advanced Security** can add, edit, and delete Keystores.

Key Alias Alias to the key in the keystore that contains the private key used to connect to the Web Service host securely. The key must be in

Field	Description
	the keystore specified in the “Keystore Alias” field. This field applies to the HTTPS transport type only.

Show Advanced Options - WS-Security properties are used by the SOAP processor to provide security information in the WS-Security header of the SOAP message.

Security Credentials

User Name	Name to include with the Username Token, if the Web Service's security policy requires one.
Password	The password to include with the UsernameToken (must be plain text).

Keystore / Truststore

Keystore Alias	The alias for the keystore, which contains private keys and certificates associated with those private keys. You can also add a new Keystore from this field.
-----------------------	---

Note: Users who have the Administer permission under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Advanced Security** can add, edit, and delete Keystores.

Key Alias	The text identifier for the private key associated with the “Keystore Alias”.
------------------	---

Truststore Alias	The alias for the truststore, which contains the trusted root of a certificate or signing authority (CA). You can also add a new Truststore from this field.
-------------------------	--

Note: Users who have the Administer permission under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Advanced Security** can add, edit, and delete Truststores.

Partner Certificate Alias	The file that contains the partner's self-signed certificate. You can also add a new Partner Certificate from this field.
----------------------------------	---

Note: Users who have the Administer permission under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Advanced Security** can add, edit, and delete Partner Certificates.

Timestamp

Field	Description
Timestamp Precision	Whether the timestamp placed in the Timestamp element of the security header of an outbound message is precise to seconds or milliseconds. If the precision is set to milliseconds, the timestamp format yyyy-MM-dd'T'HH:mm:ss:SSS'Z' is used. If the precision is set to seconds, the timestamp format yyyy-MM-dd'T'HH:mm:ss'Z' is used.
Timestamp TTL	The time-to-live value for an outbound message in seconds. This value is used to set the expiry time in the Timestamp element of outbound messages. The timestamp precision value is used only when WS-Security is implemented through a WS-Policy.
Timestamp Max Skew	The maximum number of seconds that the Web Services client and host clocks can differ so that the timestamp expiry validation does not fail. The timestamp precision value is used only when WS-Security is implemented through a WS-Policy. The inbound SOAP message is validated only if the creation timestamp of the message is less than the sum of the timestamp maximum skew value and the current system clock time.

Amazon Simple Storage Service (S3)

Field	Description
Server URL	Provide the login endpoint to initiate communication with the SaaS provider. This is the native provider endpoint target for the connection configuration.
Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. Specify a value other than 0 to avoid using a socket with no timeout. Default: 30000 ms.
Signing Algorithm	Explicitly specify the signing algorithm, for example, HMAC-SHA1 Signatures used to sign the message.
Access Key	This is a username. It is an alphanumeric text string that uniquely identifies the user who owns the account. No two accounts can have the same Access Key.
Secret Key	This key plays the role of a password. It is called secret because it is assumed to be known only by the owner. When you type the secret key, it is displayed as an asterisk or dots.

Field	Description
Region	An area specific value.

Amazon Simple Queue Service (SQS)

Field	Description
Server URL	Provide the login endpoint to initiate communication with the SaaS provider. This is the native provider endpoint target for the connection configuration.
Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. Specify a value other than 0 to avoid using a socket with no timeout. Default: 30000 ms.
Signing Algorithm	Explicitly specify the signing algorithm, for example, HMAC-SHA1 Signatures used to sign the message.
Access Key	This is a username. It is an alphanumeric text string that uniquely identifies the user who owns the account. No two accounts can have the same Access Key.
Secret Key	This key plays the role of a password. It is called secret because it is assumed to be known only by the owner. When you type the secret key, it is displayed as an asterisk or dots.
Region	An area specific value.

Salesforce CRM

Field	Description
Server URL	Provide the login endpoint to initiate communication with the SaaS provider. This is the native provider endpoint target for the connection configuration.
Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. Specify a value other than 0 to avoid using a socket with no timeout. Default: 30000 ms.

Field	Description
Username	This is the user account name on the SaaS provider that the Account will use to connect to the SaaS provider. The Account will use this credential to connect to the SaaS provider.
Password	Provide a password for the user name to initiate communication with the SaaS provider.

Salesforce Bulk Data Loader

Field	Description
Server URL	Provide the login endpoint to initiate communication with the SaaS provider. This is the native provider endpoint target for the connection configuration.
Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. Specify a value other than 0 to avoid using a socket with no timeout. Default: 30000 ms.
Username	This is the user account name on the SaaS provider that the Account will use to connect to the SaaS provider. The Account will use this credential to connect to the SaaS provider.
Password	Provide a password for the user name to initiate communication with the SaaS provider.

SuccessFactors HCM

Field	Description
Server URL	Provide the login endpoint to initiate communication with the SaaS provider. This is the native provider endpoint target for the connection configuration.
Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. Specify a value other than 0 to avoid using a socket with no timeout. Default: 30000 ms.

Field	Description
Username	This is the user account name on the SaaS provider that the Account will use to connect to the SaaS provider. The Account will use this credential to connect to the SaaS provider.
Password	Provide a password for the user name to initiate communication with the SaaS provider.
Company ID	The companyID that SuccessFactors provided, when your company registered with them.

ServiceNow Enterprise Service Management

Field	Description
Server URL	Provide the login endpoint to initiate communication with the SaaS provider. This is the native provider endpoint target for the connection configuration.
Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. Specify a value other than 0 to avoid using a socket with no timeout. Default: 30000 ms.
Username	This is the user account name on the SaaS provider that the Account will use to connect to the SaaS provider. The Account will use this credential to connect to the SaaS provider.
Password	Provide a password for the user name to initiate communication with the SaaS provider.

Strikelron Contact Verification

Field	Description
Server URL	Provide the login endpoint to initiate communication with the SaaS provider. This is the native provider endpoint target for the connection configuration.
Connection Timeout	The number of milliseconds a connection waits before canceling its attempt to connect to the resource. If you specify 0, the connection waits indefinitely. Specify a value other than 0 to avoid using a socket with no timeout. Default: 30000 ms.

Field	Description
Username	This is the user account name on the SaaS provider that the Account will use to connect to the SaaS provider. The Account will use this credential to connect to the SaaS provider.
Password	Provide a password for the user name to initiate communication with the SaaS provider.

11 Managing Operations

■ Adding or Editing Operations	54
--------------------------------------	----

Integration Cloud provides pre-configured applications. The Applications contain SaaS provider-specific information that enables you to connect to a particular SaaS provider. Further, each Application uses an Account to connect to the provider's back end and perform Operations.

Note: Users who have the required permissions under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Operations** can create, update, or delete Operations.

Each application comes with a predefined set of Operations. You can also create your own custom Operations and also edit/delete those custom Operations. This screen lists all the available Operations for a selected application including predefined Operations.

To create or edit a custom Operation

1. From the Integration Cloud navigation bar, click Applications.
2. Select an application from the list and then click **Operations**.

To use an Application, you are required to agree to the summary of terms. Click **I agree** to use the Application. Click **I do not agree** if you disagree with the summary of terms and do not want to use the Application. Click **Cancel** to go back to the **List of Applications** page.

3. From the Operations screen, click Add to add an Operation, click Edit to update an existing Operation, or click Delete to delete an existing Operation.

Adding or Editing Operations

Use the **Operations** screen to add, edit, or delete custom Operations.

To add or edit a custom Operation

1. From the Integration Cloud navigation bar, click Applications.
2. Select an application from the list and then select **Operations**.

To use an Application, you are required to agree to the summary of terms. Click **I agree** to use the Application. Click **I do not agree** if you disagree with the summary of terms and do not want to use the Application. Click **Cancel** to go back to the **List of Applications** page.

3. From the Operations screen, click Add to add a custom Operation or click Edit to update an existing custom Operation. You can edit the Operation or the Business Parameter.
4. On the Select your <...> account screen, complete the following fields. Required fields are marked with an asterisk in the screen.

Field	Description
Name	Provide a name for the custom Operation.
Description	Provide a description for the custom Operation.
Type	<p>Select the type of Operation you want to perform for the selected application.</p> <p>A Trigger is an Operation which fetches data from an application and which can be used in the Source section while creating an Integration.</p> <p>An Action is an Operation which uploads data to an application and which can be used in the Target section while creating an Integration.</p> <p>Note: You will not be allowed to change these options after the Operation is created.</p>
Select Account	Select an Account created for the Application from the drop down list. Only active or enabled Accounts are listed in the drop down list.
Select functional area	Select the Application service from the drop down list. This option is applicable only for certain applications.

5. Click Next.
6. Select the Operation to be performed and then click Next.
7. Select the business object to be processed by the Operation and then click Next. Business objects will appear only for certain applications and Operations.
8. Select the data fields to be added if any, and then click Next. Data fields will appear only for certain applications and Operations.
9. Verify the details in the Confirm Operation page and then click Finish to create the custom Operation.

Note: You will not be able to delete an Operation if it is used by an Integration.

After you click **Finish** or **Save**, if there are any Business Parameters, you will be asked to configure the Business Parameters. Currently, Business Parameters Configuration screen appears only for Upsert Operations for Salesforce.

12 Integrations

■ Creating Point-to-Point Integrations	58
■ Creating Orchestrated Integrations	60
■ Built-In Services	64
■ Reference Data	110
■ Pipeline and Signatures	111
■ Integration Details	111
■ Execution Results	114

An Integration service is an orchestration of a source and a target Operation with appropriate data mappings and transformations.

This screen lists Integrations created for cloud-based SaaS applications with other cloud-based applications and also SaaS applications with on-premise applications. The Integration *Type* shows whether the Integration is an Orchestration or a Point to Point. It also displays the Applications involved to create the Integration, when the Integration was created, and who created it. You can delete, edit, and create a new Integration from this screen.

From this page you can create point to point or orchestrated Integrations. Orchestrated Integrations will help you to join or route data across multiple SaaS and on-premise applications.

To create a new point to point Integration, click the **Add New Integration** tab, select **Synchronize two applications**, and click **OK**. To create orchestrated Integrations, select **Orchestrate two or more applications**. See "[Creating Orchestrated Integrations](#)" on page 60 for information on how to create orchestration flows.

Note: Users who have the required permissions under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Integrations** can create, update, delete, execute, or pull Integrations.

If you select an Integration and click the Integration name link under the **Name** column for that Integration, the Integration Details page will appear. The **Execution Results** link in the **My Integrations** page allows you to view the audit trail of all the executions that happened in a stage.

Creating Point-to-Point Integrations

Integration Cloud enables you to integrate your cloud-based Software as a Service (SaaS) applications with other cloud-based SaaS applications. It also integrates your SaaS applications with on-premise applications.

Integration between two cloud providers includes the following steps:

- Invoking a source Operation on an application, which fetches data from it
- Invoking a target Operation on an application, which uploads data into it
- Filtering the data fetched from an application, before it is passed on to the target application
- Mapping the data fetched from an application, to the structure needed by the target application to which you want to upload the data.

To add or edit an existing Point-to-Point Integration

1. From the Integration Cloud navigation bar, click Integrations. The “My Integrations” screen appears.

2. To edit an existing Integration, select an Integration from the “My Integrations” screen and click Edit.
3. To create a new point-to-point Integration, from the “My Integrations” screen, click “Add New Integration”, select “Synchronize two applications”, and click OK.

Note: See ["Creating Orchestrated Integrations" on page 60](#) for information on how to create an orchestrated Integration.

Note: To use an Application, you are required to agree to the summary of terms. Click **I agree** to use the Application. Click **I do not agree** if you disagree with the summary of terms and do not want to use the Application. Click **Cancel** to go back to the **List of Applications** page.

4. Provide a name and description of your Integration. Required fields are marked with an asterisk in the screen.
5. Drag and drop your applications to the Source and Target sections. You can also double-click an Application to move it to the required section.
6. Select an Account and then select a custom or a predefined Operation in both the Source and Target sections. Only active or enabled Accounts are listed in the drop down list.

Note: If you had already done the mapping for a source and target Operation, and you want to change any of the source and target Operations, all the mappings you had performed before will be removed.

7. Click Next to filter the data fetched by the application selected in the source section, before it is passed on to the application selected in the target section. Click “Load Data” to preview the data as well as view the data filters. The source Operation fetches the data and displays a sample of the data in the preview pane. Out of all the records fetched, you may want to upload only selected records. To do this, you can have a selection or a filter criteria so that you can view only a few records. A sample preview of only 12 records can be viewed to analyze the kind of data that exists in the system. After you analyze the records, you can set filters, to upload, for example only Accounts that are based out of California to the target application. After you set the filters, whenever you run the Integration, all records will be fetched from the source application, but only the filtered records will be moved to the target application after mapping and transformation.
8. Click Next to map the data fetched by the application selected in the source section, to the structure needed by the application selected in the target section.

Select **Show Mapping Hints** to display the ... icon beside relevant fields. Click on the ... icon to display mapping suggestions. You can also select a field from the source section and drag and drop it on to a relevant field in the target section. Red colored arrows indicate incorrect mappings. Select a mapping and then click **Unmap** to unmap only the selected mapping. Click **Clear All** to unmap all the mapped elements, values set to the fields, and transformers. Select a field in the target Operation table

and then choose **Set Value > Set** to set a new value of the selected field in the target Operation. You can select the field and then choose **Set Value > Clear** to clear the value.

Select **Add** from the **Set Transformer** drop down menu to display the **Select Transformer** screen. This screen allows you to transform the source Operation data, for example, concatenate two strings and map it to a single field. The *fx* icon appears after a transformer is set up correctly. You can click the *fx* icon and select **Edit** or **Delete** from the **Set Transformer** drop down menu to either modify the transformer or delete it. The **Select Transformer** screen also allows you to look up and use data from another source Operation to transform the data.

9. Click Next to review your Integration.
10. Click Save and then click Finish to create your Integration.

The new Integration appears in the “My Integrations” page.

Creating Orchestrated Integrations

Application orchestration is the process of integrating two or more applications and/or services together, to automate a process, or synchronize data in real-time. It is the coordination and arrangement of multiple services exposed as a single aggregate service. Application orchestration enables you to integrate applications and provides a way to manage and monitor your integrations centrally.

Integration Cloud supports advanced integration scenarios involving multiple application endpoints, complex routing, and orchestrations involving multiple steps. Using a graphical drag and drop tool, you can create complex, orchestrated integrations and run them in the Integration Cloud environment.

To create an orchestrated integration

1. From the Integration Cloud navigation bar, click “Integrations”. The “My Integrations” screen appears.
2. To create a new orchestration, from the “My Integrations” screen, click “Add New Integration”.
3. Select “Orchestrate two or more applications” and then click OK.

The user interface consists of a tool bar which holds all the available blocks and a workspace where you place the blocks. There is a trash can on the workspace that deletes any blocks dragged onto it. You can browse through the menu of blocks and can set up your own orchestration by plugging blocks together in the workspace. The menu of blocks comes with a large number of predefined blocks from Applications, flow control, conditions to looping structures.


4. Provide a valid name and description for the orchestration in the Start Integration and End Integration block, which is the root block for your orchestration. You can drag relevant blocks from the tool bar and attach them within the Start Integration and End Integration space.

The tool bar has a large number of blocks for common instructions. They are divided into the following categories:






- Applications
- Conditions
- Logic
- Flow Control
- Loops
- Values


Block category	Description
Applications	Displays the list of Applications available in Integration Cloud including Custom SOAP Applications.
Conditions	<p>Conditional expressions. These constructs perform different computations or actions depending on whether a specified boolean condition evaluates to true or false.</p> <p>The <i>if-then</i> block is used to evaluate a boolean condition and if the condition is true, the statements following the “then” are executed. Otherwise, the execution continues in “else” if you have selected the <i>if-then-else</i> block. You can implement error handling by using the <i>try-catch</i> block.</p>
Logic	<p>Logical operations and comparisons. The six comparison operators are: equal to, not equal to, less than, less than or equal to, greater than, greater than or equal to. Each takes two inputs and returns true or false depending on how the inputs compare with each other.</p> <p>The <i>and</i> block will return true only if both of its two inputs are also true. The <i>or</i> block will return true if either of its two inputs are true. The <i>not</i> block converts its Boolean input into its opposite.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: It is recommended not to leave an input empty.</p> </div>
Flow Control	<p>Use the <i>Transform Pipeline</i> block to make pipeline modifications. See Pipeline and Signatures for more information.</p> <p>Use the <i>Service</i> blocks (date, math, string, and so on) to specify the service that will be invoked at run time. Related services are grouped in blocks. You can sequence services and manage the flow of data among them.</p>

Block category	Description
	<p>Note: For information on the different services, see Built-In Services.</p> <p>The Reference Data block appears only if a Reference Data service is available at Integrations > Reference Data > My Reference Data. See Reference Data for more information.</p>
Loops	<p>This category executes a set of steps multiple times based on the block you have chosen. It repeats a sequence of child steps once for each element in an array that you specify. For example, if your pipeline contains an array of purchase-order line items, you could use a Loop to process each line item in the array.</p> <p>Loop requires you to specify an input array that contains the individual elements that will be used as input to one or more steps in the Loop. At run time, the Loop executes one pass of the loop for each member in the specified array. For example, if you want to execute a Loop for each line item stored in a purchase order, you would use the document list in which the order's line items are stored as the Loop's input array.</p> <p><i>while-do</i> loops repeat their bodies while the conditional expression you have provided evaluates to true. <i>do-until</i> loops are similar except that they repeat their bodies until some condition is true. The <i>for-each-do</i> block traverses items in a collection. Unlike other for loop constructs, for-each loops usually maintain no explicit counter: they essentially say "do this to everything in this set", rather than "do this x times".</p> <p>A Loop takes as input an array variable that is in the pipeline. It loops over the members of an input array, executing its child steps each time through the loop. For example, if you have a service that takes a string as input and a string list in the pipeline, use Loops to invoke the service one time for each string in the string list. You identify a single array variable to use as input when you set the properties for the Loop. You can also designate a single variable for the output. Loop collects an output value each time it runs through the loop and creates an output array that contains the collected output values.</p>
Values	<p>You can type a text value, select a field on which to build an expression, or select a block with no inputs.</p>

5. Click "Applications". The list of supported Applications appears.
6. Drag and drop an Application to the Start/End Integration block.
7. To select the Account and Operation for the Application, click 

The following table depicts the block interactions:

Button Image	Block Name	Action
	All blocks.	Add or edit the description of a block. Comments for all blocks.
	Application	Map the input of the operation from the Pipeline and also map the output of the operation into the pipeline.
	Transform Pipeline	Make pipeline modifications. Edit data mapping/Set Transformer/Set a value/Remove a field/Restore a field/Add a field/Clear all mappings.
	Application	Select an Account and an Operation for the Application.
	Application/Service	The block is not configured. Select an Account and an Operation for the Application or select a Service.

8. Create the orchestration using the available constructs by inserting the blocks, setting properties, declaring the input and output parameters, and mapping the pipeline data.
9. To map the Pipeline Input to the Input Signature, click .
10. Map the Output Signature to the Pipeline Output in the pipeline data window and then click Finish.
11. The Transform Pipeline block adjusts the pipeline at any point in the orchestration and makes pipeline modifications. Within this step, you can drop or remove an existing pipeline input field, (once you drop a field from the pipeline, it is no longer available to subsequent services in the flow), restore the dropped field, add a field, set a new value or modify the existing value of a selected field, map selected fields, remove the selected map between the fields, or perform value transformations by inserting transformers.
12. Click "Save" to save your Integration.

The new Integration will appear in the **My Integrations** page. You can click on the Integration link in the "My Integrations" page to view the details.

Built-In Services

This section describes the services provided with Integration Cloud. Services are method-like units of logic that clients can invoke. You can use services to carry out work such as extracting data from documents, interacting with back-end resources, for example, submitting a query to a database.

Integration Cloud has an extensive library of built-in services for performing common integration tasks such as transforming data values, performing simple mathematical operations, and so on. Related services are grouped in blocks. Input and output parameters are the names and types of fields that the service requires as input and generates as output. These parameters are also collectively referred to as a signature.

Date

The Date elements can be used to generate and format date values.

Pattern String Symbols - Many of the date elements require you to specify pattern strings describing the data's current format and/or the format to which you want it converted. For elements that require a pattern string, use the symbols in the following table to describe the format of your data. For example, to describe a date in the January 15, 1999 format, you would use the pattern string `MMMM dd, yyyy`. To describe the format `01/15/99`, you would use the pattern string `MM/dd/yy`. For more information about these pattern string symbols, see the Oracle Java API documentation for the `SimpleDateFormat` class.

Symbol	Meaning	Presentation	Example
G	era designator	Text	AD
y	year	Number	1996 or 96
M	month in year	Text or Number	July or Jul or 07
d	day in month	Number	10
h	hour in am/pm (1-12)	Number	12
H	hour in day (0-23)	Number	0
m	minute in hour	Number	30
s	second in minute	Number	55

Symbol	Meaning	Presentation	Example
S	millisecond	Number	978
E	day in week	Text	Tuesday or Tue
D	day in year	Number	189
F	day of week in month	Number	2 (2nd Wed in July)
w	week in year	Number	27
W	week in month	Number	2
a	am/pm marker	Text	PM
k	hour in day (1-24)	Number	24
K	hour in am/pm (0-11)	Number	0
z	time zone	Text	Pacific Standard Time or PST or GMT-08:00
Z	RFC 822 time zone (JVM 1.4 or later)	Number	-0800 (offset from GMT/UT)
'	escape for text	Delimiter	
''	single quote	Literal	'

Time Zones - When working with date elements, you can specify time zones. The Earth is divided into 24 standard time zones, one for every 15 degrees of longitude. Using the time zone including Greenwich, England (known as Greenwich Mean Time, or GMT) as the starting point, the time is increased by an hour for each time zone east of Greenwich and decreases by an hour for each time zone west of Greenwich. The time difference between a time zone and the time zone including Greenwich, England (GMT) is referred to as the *raw offset*.

The following table identifies the different time zones for the Earth and the raw offset for each zone from Greenwich, England. The effects of daylight savings time are ignored in this table.

Note: Greenwich Mean Time (GMT) is also known as Universal Time (UT).

ID	Raw Offset	Name
MIT	-11	Midway Islands Time
HST	-10	Hawaii Standard Time
AST	-9	Alaska Standard Time
PST	-8	Pacific Standard Time
PNT	-7	Phoenix Standard Time
MST	-7	Mountain Standard Time
CST	-6	Central Standard Time
EST	-5	Eastern Standard Time
IET	-5	Indiana Eastern Standard Time
PRT	-4	Puerto Rico and U.S. Virgin Islands Time
CNT	-3.5	Canada Newfoundland Time
AGT	-3	Argentina Standard Time
BET	-3	Brazil Eastern Time
GMT	0	Greenwich Mean Time
ECT	+1	European Central Time
CAT	+2	Central Africa Time
EET	+2	Eastern European Time
ART	+2	(Arabic) Egypt Standard Time
EAT	+3	Eastern African Time
MET	+3.5	Middle East Time

ID	Raw Offset	Name
NET	+4	Near East Time
PLT	+5	Pakistan Lahore Time
IST	+5.5	India Standard Time
BST	+6	Bangladesh Standard Time
VST	+7	Vietnam Standard Time
CTT	+8	China Taiwan Time
JST	+9	Japan Standard Time
ACT	+9.5	Australian Central Time
AET	+10	Australian Eastern Time
SST	+11	Solomon Standard Time
NST	+12	New Zealand Standard Time

Examples - You can specify *timezone* input parameters in the following formats:

- As a full name. For example:

`Asia/Tokyo` `America/Los_Angeles`

You can use the `java.util.TimeZone.getAvailableIDs()` method to obtain a list of the valid full name time zone IDs that your JVM version supports.

- As a custom time zone ID, in the format `GMT[+ | -]hh[[:]mm]`. For example:

`GMT+2:00` All time zones 2 hours east of Greenwich (that is, Central Africa Time, Eastern European Time, and Egypt Standard Time)

`GMT-3:00` All time zones 3 hours west of Greenwich (that is, Argentina Standard Time and Brazil Eastern Time)

`GMT+9:30` All time zones 9.5 hours east of Greenwich (that is, Australian Central Time)

- As a three-letter abbreviation from the table above. For example:

PST Pacific Standard Time

Note: Because some three-letter abbreviations can represent multiple time zones, for example, "CST" could represent both U.S. "Central Standard Time" and "China Standard Time", all abbreviations are deprecated. Use the full name or custom time zone ID formats instead.

Notes on Invalid Dates - The dates you use with a date service must adhere to the `java.text.SimpleDateFormat` class.

If you use an invalid date with a date service, the date service automatically translates the date to a legal date. For example, if you specify `1999/02/30` as input, the date service interprets the date as `1999/03/02` (two days after `2/28/1999`).

If you use `00` for the month or day, the date service interprets `00` as the last month or day in the Gregorian calendar. For example, if you specify `00` for the month, the date service interprets it as `12`.

If the pattern `yy` is used for the year, the date service uses a 50-year moving window to interpret the value of `yy`. The date service establishes the window by subtracting 49 years from the current year and adding 50 years to the current year. For example, if you are running the webMethods Integration Server in the year 2000, the moving window would be from 1951 to 2050. The date service interprets 2-digit years as falling into this window (for example, `12` would be 2012, `95` would be 1995).

Summary of Date elements

The following Date elements are available:

Element	Description
<code>calculateDateDifference</code>	Calculates the difference between two dates and returns the result as seconds, minutes, hours, and days.
<code>dateBuild</code>	Builds a date String using the specified pattern and the specified date elements.
<code>dateTimeBuild</code>	Builds a date/time string using the specified pattern and the specified date elements.
<code>dateTimeFormat</code>	Converts date/time (represented as a String) string from one format to another.

Element	Description
<code>getCurrentDateString</code>	Returns the current date as a String in a specified format.

calculateDateDifference

Calculates the difference between two dates and returns the result as seconds, minutes, hours, and days.

Input Parameters

<i>startDate</i>	String Starting date and time.
<i>endDate</i>	String Ending date and time.
<i>startDatePattern</i>	String Format in which the <i>startDate</i> parameter is to be specified (for example, yyyyMMdd HH:mm:ss.SSS). For pattern-string notation, see the “Pattern String Symbols” section.
<i>endDatePattern</i>	String Format in which the <i>endDate</i> parameter is to be specified (for example, yyyyMMdd HH:mm:ss.SSS). For pattern-string notation, see the “Pattern String Symbols” section.

Output Parameters

<i>dateDifferenceSeconds</i>	String The difference between the <i>startingDateTime</i> and <i>endingDateTime</i> , truncated to the nearest whole number of seconds.
<i>dateDifferenceMinutes</i>	String The difference between the <i>startingDateTime</i> and <i>endingDateTime</i> , truncated to the nearest whole number of minutes.
<i>dateDifferenceHours</i>	String The difference between the <i>startingDateTime</i> and <i>endingDateTime</i> , truncated to the nearest whole number of hours.
<i>dateDifferenceDays</i>	String The difference between the <i>startingDateTime</i> and <i>endingDateTime</i> , truncated to the nearest whole number of days.

Usage Notes

Each output value represents the same date difference, but in a different scale. Do not add these values together. Make sure your subsequent flow steps use the correct output, depending on the scale required.

dateBuild

Builds a date String using the specified pattern and the specified date elements.

Input Parameters

<i>pattern</i>	String Pattern representing the format in which you want the date returned. For pattern-string notation, see the “Pattern String Symbols” section. If you do not specify <i>pattern</i> , <i>dateBuild</i> returns null. If <i>pattern</i> contains a time zone and <i>timezone</i> is not specified, the default time zone is used.
<i>year</i>	String Optional. The year expressed in <i>yyyy</i> or <i>yy</i> format (for example, <i>01</i> or <i>2001</i>). If you do not specify <i>year</i> or you specify an invalid value, <i>dateBuild</i> uses the current year.
<i>month</i>	String Optional. The month expressed as a number (for example, <i>1</i> for January, <i>2</i> for February). If you do not specify <i>month</i> or you specify an invalid value, <i>dateBuild</i> uses the current month.
<i>dayofmonth</i>	String Optional. The day of the month expressed as a number (for example, <i>1</i> for the first day of the month, <i>2</i> for the second day of the month). If you do not specify <i>dayofmonth</i> or you specify an invalid value, <i>dateBuild</i> uses the current day.
<i>timezone</i>	String Optional. Time zone in which you want the output date and time expressed. Specify a time zone code as shown in the “Time Zones” section, for example, <i>EST</i> for Eastern Standard Time. If you do not specify <i>timezone</i> , the value of the server's "user timezone" property is used. If this property has not been set, GMT is used.
<i>locale</i>	String Optional. Locale in which the date is to be expressed. For example, if <i>locale</i> is <i>en</i> (for English), the pattern <i>EEE d MMM yyyy</i> will produce <i>Friday 23 August 2002</i> , and the <i>locale</i> of <i>fr</i> (for French) will produce <i>vendredi 23 août 2002</i> .

Output Parameters

value **String** The date specified by *year* , *month* , and *dayofmonth* , in the format of *pattern* .

dateTimeBuild

Builds a date/time string using the specified pattern and the specified date elements.

Input Parameters

pattern **String** Pattern representing the format in which you want the time returned. For pattern-string notation, see the “Pattern String Symbols” section. If you do not specify *pattern* , *dateTimeBuild* returns null. If *pattern* contains a time zone and the *timezone* parameter is not set, the default time zone is used.

year **String** Optional. The year expressed in *yyyy* or *yy* format (for example, *01* or *2001*). If you do not specify *year* or you specify an invalid value, *dateTimeBuild* uses the current year.

month **String** Optional. The month expressed as a number (for example, *1* for January, *2* for February). If you do not specify *month* or you specify an invalid value, *dateTimeBuild* uses the current month.

dayofmonth **String** Optional. The day of the month expressed as a number (for example, *1* for the first day of the month, *2* for the second day of the month). If you do not specify *dayofmonth* or you specify an invalid value, *dateTimeBuild* uses the current day.

hour **String** Optional. The hour expressed as a number based on a 24-hour clock. For example, specify *0* for midnight, *2* for 2:00 A.M., and *14* for 2:00 P.M. If you do not specify *hour* or you specify an invalid value, *dateTimeBuild* uses *0* as the *hour* value.

minute **String** Optional. Minutes expressed as a number. If you do not specify *minute* or you specify an invalid value, *dateTimeBuild* uses *0* as the *minute* value.

second **String** Optional. Seconds expressed as a number. If you do not specify *second* or you specify an invalid value, *dateTimeBuild* uses *0* as the *second* value.

<i>millis</i>	String Optional. Milliseconds expressed as a number. If you do not specify <i>millis</i> or you specify an invalid value, <code>dateTimeBuild</code> uses 0 as the <i>millis</i> value.
<i>timezone</i>	String Optional. Time zone in which you want the output date and time expressed. Specify a time zone code as shown in the “Time Zones” section, for example, <code>EST</code> for Eastern Standard Time. If you do not specify <i>timezone</i> , the value of the server's "user timezone" property is used. If this property has not been set, GMT is used.
<i>locale</i>	String Optional. Locale in which the date is to be expressed. For example, if <i>locale</i> is <code>en</code> (for English), the pattern <code>EEE d MMM YYYY</code> will produce <code>Friday 23 August 2002</code> , and the <i>locale</i> of <code>fr</code> (for French) will produce <code>vendredi 23 août 2002</code> .

Output Parameters

<i>value</i>	String Date and time in format of <i>pattern</i> .
--------------	---

dateTimeFormat

Converts date/time (represented as a `String`) string from one format to another.

Input Parameters

<i>inString</i>	String Date/time that you want to convert. Important: If <i>inString</i> contains a character in the last position, that character is interpreted as 0. This can result in an inaccurate date. For information about invalid dates, see the “Notes on Invalid Dates” section.
<i>currentPattern</i>	String Pattern string that describes the format of <i>inString</i> . For pattern-string notation, see the “Pattern String Symbols” section.
<i>newPattern</i>	String Pattern string that describes the format in which you want <i>inString</i> returned. For pattern-string syntax, see the “Pattern String Symbols” section.
<i>locale</i>	String Optional. Locale in which the date is to be expressed. For example, if <i>locale</i> is <code>en</code> (for English), the pattern <code>EEE d MMM YYYY</code>

will produce `Friday 23 August 2002`, and the *locale* of `fr` (for French) will produce `vendredi 23 août 2002`.

lenient

String Optional. A flag indicating whether an exception will appear if the *inString* value does not adhere to the format specified in *currentPattern* parameter. Set to:

- `true` to perform a lenient check. This is the default.

In a lenient check, if the format of the date specified in the *inString* parameter does not match the format specified in the *currentPattern* parameter, the date in the format specified in the *currentPattern* parameter will be interpreted and returned. If the interpretation is incorrect, the service will return an invalid date.

- `false` to perform a strict check.

In a strict check, an exception will appear if the format of the date specified in the *inString* parameter does not match the format specified in the *currentPattern* parameter.

Output Parameters

value

String The date/time given by *inString*, in the format of *newPattern*.

Usage Notes

As described in the “Notes on Invalid Dates” section, if the pattern *yy* is used for the year, `dateTimeFormat` uses a 50-year moving window to interpret the value of the year. If you need to change this behavior so that the year is interpreted as 80 years before or 20 years after the current date (as described in the Java class `java.text.SimpleDateFormat`), set the server configuration parameter `watt.server.pubDateTimeFormat.javaSlidingWindow` to `true`. For information about setting configuration parameters, see *webMethods Integration Server Administrator’s Guide*.

By default, an exception will appear if no input is passed to the service. To suppress the error message and return a null value for the *value* parameter, set the server configuration parameter `watt.server.date.suppressPatternError` to `true`. For information about setting configuration parameters, see *webMethods Integration Server Administrator’s Guide*.

If *currentPattern* does not contain a time zone, the value is assumed to be in the default time zone.

If *newPattern* contains a time zone, the default time zone is used.

getCurrentDateString

Returns the current date as a String in a specified format.

Input Parameters

<i>pattern</i>	String Pattern representing the format in which you want the date returned. For pattern-string notation, see the “Pattern String Symbols” section.
<i>timezone</i>	String Optional. Time zone in which you want the output date and time expressed. Specify a time zone code as shown in the “Time Zones” section, for example, <code>EST</code> for Eastern Standard Time. If you do not specify <i>timezone</i> , the value of the server's "user timezone" property is used. If this property has not been set, GMT is used.
<i>locale</i>	String Optional. Locale in which the date is to be expressed. For example, if <i>locale</i> is <code>en</code> (for English), the pattern <code>EEE d MMM yyyy</code> will produce <code>Friday 23 August 2002</code> , and the <i>locale</i> of <code>fr</code> (for French) will produce <code>vendredi 23 août 2002</code> .

Output Parameters

<i>value</i>	String Current date in the format specified by <i>pattern</i> .
--------------	--

Documents

The following document elements are available:

Element	Description
findDocuments	Searches a set of documents for entries matching a set of Criteria.

Summary of Document elements

findDocuments

Searches a set of documents for entries matching a set of criteria.

Input Parameters

- documents* **Document List** Set of documents from which the documents meeting the retrieve criteria are to be returned.
- matchCriteria* **Document** Criteria on which the documents in the `documents` parameter are to be matched. Valid values for `matchCriteria` parameters are:
- Path:** Name of the element in `documentList` whose value provides the value for the search text. The value for `key` can be a path expression. For example, "Family/Children[0]/BirthDate" retrieves the birthday of the first child from the input `Family` document list.
 - CompareValueAs:** Optional. Allowed values are string, numeric, and datetime. The default value is string.
 - datePattern:** Optional. Pattern will be considered only if `CompareValueAs` is of type datetime.
- joins* List of join criteria. Each join criteria consists of:
- Value:** Optional. Allowed values are string, numeric, and datetime. The default value is string.
 - Operator:** Allowed values are equals, doesNotEqual, greaterThan, greaterThanEqual, lessThan, lessThanEqual, equalsIgnoreCase, contains, doesNotContain, beginsWith, doesNotBeginWith, endsWith, doesNotEndWith.
 - joinType:** Specifies the way two joins can be linked. Values are "and" or "or".

Output Parameters

- resultdocuments* **Document List** List of documents which are matching the retrieve criteria.

Math

You can use the math elements to perform mathematical operations on string-based numeric values. Services that operate on integer values use Java's long data type (64-bit, two's complement). Services that operate on float values use Java's double data type (64-bit IEEE 754). If extremely precise calculations are critical to your application, you should write your own Java services to perform math functions. The following elements are available in this folder:

Element	Description
absoluteValue	Returns the absolute value of the input number.
addFloatList	Adds a list of floating point numbers (represented in a string list) and returns the sum.
addFloats	Adds one floating point number (represented as a String) to another and returns the sum.
addIntList	Adds a list of integers (represented in a String list) and returns the sum.
addInts	Adds one integer (represented as a String) to another and returns the sum.
divideFloats	Divides one floating point number (represented as a String) by another ($num1/num2$) and returns the quotient.
divideInts	Divides one integer (represented as a String) by another ($num1/num2$) and returns the quotient.
max	Returns the largest number from a list of numbers.
multiplyFloatList	Multiplies a list of floating point numbers (represented in a String list) and returns the product.
multiplyFloats	Multiplies one floating point number (represented as String) by another and returns the product.
multiplyIntList	Multiplies a list of integers (represented in a String list) and returns the product.

Element	Description
<code>multiplyInts</code>	Multiplies one integer (represented as a String) by another and returns the product.
<code>randomDouble</code>	Returns the next pseudorandom, uniformly distributed double between 0.0 and 1.0.
<code>roundNumber</code>	Returns a rounded number.
<code>subtractFloats</code>	Subtracts one floating point number (represented as a String) from another and returns the difference.
<code>subtractInts</code>	Subtracts one integer (represented as a String) from another and returns the difference.

Summary of Math elements

absoluteValue

Returns the absolute value of the input number.

Input Parameters

num **String** Number whose absolute value is to be returned.

Output Parameters

positiveNumber **String** Absolute value of the input number.

addFloatList

Adds a list of floating point numbers (represented in a string list) and returns the sum.

Input Parameters

numList **String List** Numbers (floating point numbers represented in a string list) to add.

Output Parameters

value **String** Sum of the numbers in *numList* . If a sum cannot be produced, *value* contains one of the following:

Value	Description
Infinity	The computation produces a positive value that overflows the representable range of a float type.
-Infinity	The computation produces a negative value that overflows the representable range of a float type.
0.0	The computation produces a value that underflows the representable range of a float type (for example, adding a number to infinity).
NaN	The computation produces a value that cannot be represented as a number (for example, any operation that uses NaN as input, such as $10.0 + \text{NaN} = \text{NaN}$).

Usage Notes

Make sure the strings that are passed to the service in *numList* are in a locale-neutral format (that is, using the pattern `-####.##`). Passing locally formatted strings may result in unexpected results. For example, calling `addFloats` in a German locale with the arguments `1,23` and `2,34` will result in the value `357`, not `3.57` or `3,57`.

addFloats

Adds one floating point number (represented as a String) to another and returns the sum.

Input Parameters

num1 **String** Number to add.

num2 **String** Number to add.

precision **String** Optional. Number of decimal places to which the sum will be rounded. The default value is null.

The *precision* parameter, if specified, will override the behavior set by the `watt.server.math.floatOperation.mode` property. For information

about the `watt.server.math.floatOperation.mode` property, see the *webMethods Integration Server Administrator's Guide*.

Output Parameters

value **String** Sum of the numbers in *num1* and *num2*. If a sum cannot be produced, *value* contains one of the following:

Value	Description
Infinity	The computation produces a positive value that overflows the representable range of a float type.
-Infinity	The computation produces a negative value that overflows the representable range of a float type.
0.0	The computation produces a value that underflows the representable range of a float type (for example, adding a number to infinity).
NaN	The computation produces a value that cannot be represented as a number (for example, any operation that uses NaN as input, such as $10.0 + \text{NaN} = \text{NaN}$).

Usage Notes

Make sure the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern `-###.##`). Passing locally formatted strings may result in unexpected results. For example, calling `addFloats` in a German locale with the arguments `1,23` and `2,34` will result in the value `357`, not `3.57` or `3,57`.

Use the `watt.server.math.floatOperation.mode` property to specify whether the `pub.math:addFloats` service return the exact result of an operation involving two floating point numbers, the result as calculated by the JVM, or the result based on a fixed number of decimal places. For information about the `watt.server.math.floatOperation.mode` property, see the *webMethods Integration Server Administrator's Guide*.

addIntList

Adds a list of integers (represented in a String list) and returns the sum.

Input Parameters

numList **String List** Numbers (integers represented as Strings) to add.

Output Parameters

value **String** Sum of the numbers in *numList*.

Usage Notes

Make sure the strings that are passed to the service in *numList* are in a locale-neutral format (that is, using the pattern -####.##). Passing locally formatted strings may result in unexpected results. For example, calling [addFloats](#) in a German locale with the arguments 1,23 and 2,34 will result in the value 357, not 3.57 or 3,57.

addInts

Adds one integer (represented as a String) to another and returns the sum.

Input Parameters

num1 **String** Number (integer represented as a String) to add.

num2 **String** Number (integer represented as a String) to add.

Output Parameters

value **String** Sum of *num1* and *num2*.

Usage Notes

Ensure that the result of your calculation is less than 64 bits in width (the maximum width for the long data type). If the result exceeds this limit, it will generate a data overflow.

Ensure that the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern -####.##). Passing locally formatted strings may result in unexpected results. For example, calling [addFloats](#) in a German locale with the arguments 1,23 and 2,34 will result in the value 357, not 3.57 or 3,57.

divideFloats

Divides one floating point number (represented as a String) by another (*num1/num2*) and returns the quotient.

Input Parameters

num1 **String** Number (floating point number represented as a String) that is the dividend.

num2 **String** Number (floating point number represented as a String) that is the divisor.

precision **String** Optional. Number of decimal places to which the quotient will be rounded. The default value is null.

The *precision* parameter, if specified, will override the behavior set by the `watt.server.math.floatOperation.mode` property. For information about the `watt.server.math.floatOperation.mode` property, see the *webMethods Integration Server Administrator's Guide*.

Output Parameters

value **String** The quotient of *num1 / num2*. If a quotient cannot be produced, *value* contains one of the following:

Value	Description
Infinity	The computation produces a positive value that overflows the representable range of a float type.
- Infinity	The computation produces a negative value that overflows the representable range of a float type.
0.0	The computation produces a value that underflows the representable range of a float type (for example, dividing a number by infinity).
NaN	The computation produces a value that cannot be represented as a number (for example, the result of an illegal operation such as dividing zero by zero or any operation that uses NaN as input, such as $10.0 + \text{NaN} = \text{NaN}$).

Usage Notes

Make sure the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern -####.##). Passing locally formatted strings may result in unexpected results. For example, calling [addFloats](#) in a German locale with the arguments 1,23 and 2,34 will result in the value 357, not 3.57 or 3,57.

Use the `watt.server.math.floatOperation.mode` property to specify whether the `pub.math:divideFloats` service return the exact result of an operation involving two floating point numbers, the result as calculated by the JVM, or the result based on a fixed number of decimal places. For information about the `watt.server.math.floatOperation.mode` property, see the *webMethods Integration Server Administrator's Guide*.

divideInts

Divides one integer (represented as a String) by another (*num1/num2*) and returns the quotient.

Input Parameters

num1 **String** Number (integer represented as a String) that is the dividend.

num2 **String** Number (integer represented as a String) that is the divisor.

Output Parameters

value **String** The quotient of *num1 / num2*.

Usage Notes

Make sure the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern -####.##). Passing locally formatted strings may result in unexpected results. For example, calling [addFloats](#) in a German locale with the arguments 1,23 and 2,34 will result in the value 357, not 3.57 or 3,57.

max

Returns the largest number from a list of numbers.

Input Parameters

numList **String List** List of numbers from which the largest number is to be returned.

Output Parameters

maxValue **String** Largest number from the list of numbers.

multiplyFloatList

Multiplies a list of floating point numbers (represented in a String list) and returns the product.

Input Parameters

numList **String List** Numbers (floating point numbers represented as Strings) to multiply.

Output Parameters

value **String** Product of the numbers in *numlist* . If a product cannot be produced, *value* contains one of the following:

Value	Description
Infinity	The computation produces a positive value that overflows the representable range of a float type.
- Infinity	The computation produces a negative value that overflows the representable range of a float type.
0.0	The computation produces a value that underflows the representable range of a float type (for example, multiplying a number by infinity).
NaN	The computation produces a value that cannot be represented as a number (for example, the result of an illegal operation such as multiplying zero by zero or any operation that uses NaN as input, such as $10.0 + \text{NaN} = \text{NaN}$).

Usage Notes

Make sure the strings that are passed to the service in *numList* are in a locale-neutral format (that is, using the pattern -####.##). Passing locally formatted strings may result in unexpected results. For example, calling `addFloats` in a German locale with the arguments 1,23 and 2,34 will result in the value 357, not 3.57 or 3,57.

multiplyFloats

Multiplies one floating point number (represented as String) by another and returns the product.

Input Parameters

<i>num1</i>	String Number (floating point number represented as a String) to multiply.
<i>num2</i>	String Number (floating point number represented as a String) to multiply.
<i>precision</i>	String Optional. Number of decimal places to which the product will be rounded. The default value is null. The <i>precision</i> parameter, if specified, will override the behavior set by the <code>watt.server.math.floatOperation.mode</code> property. For information about the <code>watt.server.math.floatOperation.mode</code> property, see the <i>webMethods Integration Server Administrator's Guide</i> .

Output Parameters

value **String** Product of the numeric values of *num1* and *num2*. If a product cannot be produced, *value* contains one of the following:

<u>Value</u>	<u>Description</u>
Infinity	The computation produces a positive value that overflows the representable range of a float type.
- Infinity	The computation produces a negative value that overflows the representable range of a float type.

0.0	The computation produces a value that underflows the representable range of a float type (for example, multiplying a number by infinity).
NaN	The computation produces a value that cannot be represented as a number (for example, the result of an illegal operation such as multiplying zero by zero or any operation that uses NaN as input, such as $10.0 + \text{NaN} = \text{NaN}$).

Usage Notes

Make sure the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern `-####.##`). Passing locally formatted strings may result in unexpected results. For example, calling `addFloats` in a German locale with the arguments `1,23` and `2,34` will result in the value `357`, not `3.57` or `3,57`.

Use the `watt.server.math.floatOperation.mode` property to specify whether the `multiplyFloats` service return the exact result of an operation involving two floating point numbers, the result as calculated by the JVM, or the result based on a fixed number of decimal places. See *webMethods Integration Server Administrator's Guide* for more information about the `watt.server.math.floatOperation.mode` property.

multiplyIntList

Multiplies a list of integers (represented in a String list) and returns the product.

Input Parameters

numList **String List** Numbers (floating point numbers represented as Strings) to multiply.

Output Parameters

value **String** Product of the numbers in *numList*.

Usage Notes

Make sure the result of your calculation is less than 64 bits in width (the maximum width for the long data type). If the result exceeds this limit, it will generate a data overflow.

Make sure the strings that are passed to the service in *numList* are in a locale-neutral format (that is, using the pattern `-####.##`). Passing locally formatted strings may

result in unexpected results. For example, calling `addFloats` in a German locale with the arguments `1,23` and `2,34` will result in the value `357`, not `3.57` or `3,57`.

multiplyInts

Multiplies one integer (represented as a String) by another and returns the product.

Input Parameters

num1 **String** Number (integer represented as a String) to multiply.

num2 **String** Number (integer represented as a String) to multiply.

Output Parameters

value **String** Product of *num1* and *num2*.

Usage Notes

Make sure the result of your calculation is less than 64 bits in width (the maximum width for the long data type). If the result exceeds this limit, it will generate a data overflow.

Make sure the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern `-####.##`). Passing locally formatted strings may result in unexpected results. For example, calling `addFloats` in a German locale with the arguments `1,23` and `2,34` will result in the value `357`, not `3.57` or `3,57`.

randomDouble

Returns the next pseudorandom, uniformly distributed double between 0.0 and 1.0.

Random number generators are often referred to as pseudorandom number generators because the numbers produced tend to repeat themselves over time.

Input Parameters

None.

Output Parameters

number **String** Generated random number.

roundNumber

Returns a rounded number.

Input Parameters

<i>num</i>	String Number to be rounded.
<i>numberOfDigits</i>	String Specifies the number of digits to which you want to round the number.
<i>roundingMode</i>	String Optional. Specifies the rounding method. Valid values for the <i>roundingMode</i> parameter are RoundHalfUp, RoundUp, RoundDown, RoundCeiling, RoundFloor, RoundHalfDown, and RoundHalfEven. The default value is RoundHalfUp.

Output Parameters

<i>roundedNumber</i>	String The rounded number.
----------------------	-----------------------------------

subtractFloats

Subtracts one floating point number (represented as a String) from another and returns the difference.

Input Parameters

<i>num1</i>	String Number (floating point number represented as a String).
<i>num2</i>	String Number (floating point number represented as a String) to subtract from <i>num1</i> .
<i>precision</i>	String Optional. Number of decimal places to which the difference will be rounded. The default value is null. The <i>precision</i> parameter, if specified, will override the behavior set by the <code>watt.server.math.floatOperation.mode</code> property. For information about the <code>watt.server.math.floatOperation.mode</code> property, see the <i>webMethods Integration Server Administrator's Guide</i> .

Output Parameters

value **String** Difference of *num1* - *num2* . If a difference cannot be produced, *value* contains one of the following:

Value	Description
Infinity	The computation produces a positive value that overflows the representable range of a float type.
-Infinity	The computation produces a negative value that overflows the representable range of a float type.
0.0	The computation produces a value that underflows the representable range of a float type (for example, subtracting a number from infinity).
NaN	The computation produces a value that cannot be represented as a number (for example, the result of an illegal operation such as multiplying zero by zero or any operation that uses NaN as input, such as $10.0 - \text{NaN} = \text{NaN}$).

Usage Notes

Make sure the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern -####.##). Passing locally formatted strings may result in unexpected results. For example, calling [addFloats](#) in a German locale with the arguments 1,23 and 2,34 will result in the value 357, not 3.57 or 3,57.

Use the `watt.server.math.floatOperation.mode` property to specify whether the `pub.math.subtractFloats` service return the exact result of an operation involving two floating point numbers, the result as calculated by the JVM, or the result based on a fixed number of decimal places. For more information about the `watt.server.math.floatOperation.mode` property, see *webMethods Integration Server Administrator's Guide*

subtractInts

Subtracts one integer (represented as a String) from another and returns the difference.

Input Parameters

num1 **String** Number (integer represented as a String).

num2 **String** Number (integer represented as a String) to subtract from *num1* .

Output Parameters

value **String** Difference of *num1* - *num2* .

Usage Notes

Make sure the result of your calculation is less than 64 bits in width (the maximum width for the long data type). If the result exceeds this limit, it will generate a data overflow.

Make sure the strings that are passed to the service in *num1* and *num2* are in a locale-neutral format (that is, using the pattern -####.##). Passing locally formatted strings may result in unexpected results. For example, calling [addFloats](#) in a German locale with the arguments 1,23 and 2,34 will result in the value 357, not 3.57 or 3,57.

String

You can use the String elements to perform string manipulation and substitution operations. The following elements are available:

Element	Description
base64Decode	Decodes a Base-64 encoded string into a sequence of bytes.
base64Encode	Converts a sequence of bytes into a Base64-encoded String.
bytesToString	Converts a sequence of bytes to a String.
concat	Concatenates two strings.
indexOf	Returns the index of the first occurrence of a sequence of characters in a string.
length	Returns the length of a string.

Element	Description
lookupDictionary	Looks up a given key in a hash table and returns the string to which that key is mapped.
makeString	Builds a single string by concatenating the elements of a String List.
messageFormat	Formats an array of strings into a given message pattern.
numericFormat	Formats a number into a given numeric pattern.
padLeft	Pads a string to a specified length by adding pad characters to the beginning of the string.
padRight	Pads a string to a specified length by adding pad characters to the end of the string.
replace	Replaces all occurrences of a specified substring with a substitute string.
stringToBytes	Converts a string to a byte array.
substring	Returns a substring of a given string.
tokenize	Tokenizes a string using specified delimiter characters and generates a String List from the resulting tokens.
toLowerCase	Converts all characters in a given string to lowercase.
toUpperCase	Converts all characters in a given string to uppercase.
trim	Trims leading and trailing white space from a given string.
URLDecode	Decodes a URL-encoded string.
URLEncode	URL-encodes a string.

Summary of String elements

base64Decode

Decodes a Base-64 encoded string into a sequence of bytes.

Input Parameters

string **String** A Base64-encoded String to decode into bytes.

Output Parameters

value **byte[]** The sequence of bytes decoded from the Base64-encoded String.

encoding **String** Optional. Specifies the encoding method. Default value is ASCII.

base64Encode

Converts a sequence of bytes into a Base64-encoded String.

Input Parameters

bytes **byte[]** Sequence of bytes to encode into a Base64-encoded String.

useNewLine **String** Optional. Flag indicating whether to retain or remove the line breaks. Set to:

- `true` to retain the line breaks. This is the default.
- `false` to remove the line breaks.

encoding **String** Optional. Specifies the encoding method. Default value is ASCII.

Output Parameters

value **String** Base64-encoded String encoded from the sequence of bytes.

Usage Notes

By default, the `base64Encode` service inserts line breaks after 76 characters of data, which is not the canonical lexical form expected by implementations such as MTOM. You can use the `useNewLine` parameter to remove the line breaks. For more information about MTOM implementations, see the *Web Services Developer's Guide*.

bytesToString

Converts a sequence of bytes to a String.

Input Parameters

bytes **byte[]** Sequence of bytes to convert to a String.

encoding **String** Optional. Name of a registered, IANA character set (for example, `ISO-8859-1`). If you specify an unsupported encoding, the system throws an exception.

To use the default encoding, set *encoding* to `autoDetect`.

ignoreBOMChars **String** Optional. Flag indicating whether or not the byte order mark (BOM) characters in the input sequence of bytes are removed before converting the byte array to string. Set to:

- `true` to remove the byte order mark (BOM) characters before converting the input sequence of bytes to string, if the byte array contains BOM characters.
- `false` to include the byte order mark (BOM) characters while converting the input sequence of bytes to string. The default is `false`.

Output Parameters

string **String** String representation of the contents of *bytes* .

concat

Concatenates two strings.

Input Parameters

inString1 **String** String to which you want to concatenate another string.

inString2 **String** String to concatenate to *inString1* .

Output Parameters

value **String** Result of concatenating *inString1* with *inString2* (*inString1* + *inString2*).

indexOf

Returns the index of the first occurrence of a sequence of characters in a string.

Input Parameters

inString **String** String in which you want to locate a sequence of characters.

subString **String** Sequence of characters to locate.

fromIndex **String** Optional. Index of *inString* from which to start the search. If no value is specified, this parameter contains 0 to indicate the beginning of the string.

Output Parameters

value **String** Index of the first occurrence of *subString* in *inString* . If no occurrence is found, this parameter contains -1.

length

Returns the length of a string.

Input Parameters

inString **String** String whose length you want to discover.

Output Parameters

value **String** The number of characters in *inString* .

lookupDictionary

Looks up a given key in a hash table and returns the string to which that key is mapped.

Input Parameters

hashtable **java.util.Hashtable** Hash table that uses String objects for keys and values.

key **String** Key in *hashtable* whose value you want to retrieve.

Note: The key is case sensitive.

Output Parameters

value **String** Value of the string to which *key* is mapped. If the requested key in *hashtable* is null or if *key* is not mapped to any value in *hashtable* , the service returns null.

makeString

Builds a single string by concatenating the elements of a String List.

Input Parameters

<i>elementList</i>	String List Strings to concatenate.
<i>separator</i>	String String to insert between each non-null element in <i>elementList</i> .

Output Parameters

<i>value</i>	String Result from concatenating the strings in <i>elementList</i> . Strings are separated by the characters specified in <i>separator</i> .
--------------	---

messageFormat

Formats an array of strings into a given message pattern.

Input Parameters

<i>pattern</i>	String Message that includes "placeholders" where elements from <i>argumentList</i> are to be inserted. The message can contain any sequence of characters. Use the { <i>n</i> } placeholder to insert elements from <i>argumentList</i> , where <i>n</i> is the index of the element that you want to insert. For example, the following pattern string inserts elements 0 and 1 into the message: Test results: {0} items passed, {1} items failed. Note: Do not use any characters except digits for <i>n</i> .
<i>argumentList</i>	String List Optional. List of strings to use to populate <i>pattern</i> . If <i>argumentList</i> is not supplied, the service will not replace placeholders in <i>pattern</i> with actual values.

Output Parameters

<i>value</i>	String Result from substituting <i>argumentList</i> into <i>pattern</i> . If <i>pattern</i> is empty or null, this parameter is null.
--------------	--

numericFormat

Formats a number into a given numeric pattern.

Input Parameters

<i>num</i>	String The number to format.
<i>pattern</i>	String A pattern string that describes the way in which <i>num</i> is to be formatted:

<u>This symbol...</u>	<u>Indicates...</u>
0	A digit.
#	A digit. Leading zeroes will not be shown.
.	A placeholder for a decimal separator.
,	A placeholder for a grouping separator.
;	A separation in format.
-	The default negative prefix.
%	That <i>num</i> will be multiplied by 100 and shown as a percentage.
x	Any character used as a prefix or suffix (for example, ¤, \$).
'	That special characters are to be used as literals in a prefix or suffix. Enclose the special characters within " (for example, '#').

The following are examples of pattern strings:

<u>Pattern</u>	<u>Description</u>
----------------	--------------------

<code>#,###</code>	Use commas to separate into groups of three digits. The pound sign denotes a digit and the comma is a placeholder for the grouping separator.
<code>#,####</code>	Use commas to separate into groups of four digits.
<code>\$.00</code>	Show digits before the decimal point as needed and exactly two digits after the decimal point. Prefix with the \$ character.
<code>'#'#.0</code>	Show digits before the decimal point as needed and exactly one digit after the decimal point. Prefix with the # character. The first character in a pattern is the dollar sign (\$). The pound sign denotes a digit and the period is a placeholder for decimal separator.

Output Parameters

value **String** *num* formatted according to *pattern*. If *pattern* is an empty (not null) string, the default pattern of comma separators is used and the number of digits after the decimal point remains unchanged.

padLeft

Pads a string to a specified length by adding pad characters to the beginning of the string.

Input Parameters

inString **String** String that you want to pad.

padString **String** Characters to use to pad *inString*.

length **String** Total length of the resulting string, including pad characters.

Output Parameters

value **String** Contents of *inString* preceded by as many pad characters as needed so that the total length of the string equals *length* .

Usage Notes

If *padString* is longer than one character and does not fit exactly into the resulting string, the beginning of *padString* is aligned with the beginning of the resulting string. For example, suppose *inString* equals `shipped` and *padString* equals `x9y`.

If <i>length</i> equals...	Then <i>value</i> will contain...
7	<code>shipped</code>
10	<code>x9yshipped</code>
12	<code>x9x9yshipped</code>

If *inString* is longer than *length* characters, only the last *length* characters from *inString* are returned. For example, if *inString* equals `acct1234` and *length* equals 4, *value* will contain `1234`.

padRight

Pads a string to a specified length by adding pad characters to the end of the string.

Input Parameters

inString **String** String that you want to pad.

padString **String** Characters to use to pad *inString* .

length **String** Total length of the resulting string, including pad characters.

Output Parameters

value **String** Contents of *inString* followed by as many pad characters as needed so that the total length of the string equals *length* .

Usage Notes

If *padString* is longer than one character and does not fit exactly into the resulting string, the end of *padString* is aligned with the end of the resulting string. For example, suppose *inString* equals `shipped` and *padString* equals `x9y`.

If <i>length</i> equals...	Then <i>value</i> will contain...
7	<code>shipped</code>
10	<code>shippedx9y</code>
12	<code>shippedx9y9y</code>

If *inString* is longer than *length* characters, only the first *length* characters from *inString* are returned. For example, if *inString* equals `1234acct` and *length* equals 4, value will contain `1234`.

replace

Replaces all occurrences of a specified substring with a substitute string.

Input Parameters

<i>inString</i>	String String containing the substring to replace.
<i>searchString</i>	String Substring to replace within <i>inString</i> .
<i>replaceString</i>	String Character sequence that will replace <i>searchString</i> . If this parameter is null or empty, the service removes all occurrences of <i>searchString</i> from <i>inString</i> .
<i>useRegex</i>	String Optional. Flag indicating whether <i>searchString</i> is a regular expression. When regular expressions are used to specify a search string, <i>replaceString</i> may also contain interpolation variables (for example, "\$1") that match parenthetical subexpressions in <i>searchString</i> . Set to: <ul style="list-style-type: none"> ■ <code>true</code> to indicate that <i>searchString</i> is a regular expression. ■ <code>false</code> to indicate that <i>searchString</i> is not a regular expression. This is the default.

Output Parameters

value **String** Contents of *inString* with replacements made.

stringToBytes

Converts a string to a byte array.

Input Parameters

string **String** String to convert to a byte[].

encoding **String** Optional. Name of a registered, IANA character set that specifies the encoding to use when converting the String to an array of bytes (for example: ISO-8859-1).

To use the default encoding, set this value to `autoDetect`. If you specify an unsupported encoding, an exception will be thrown.

Output Parameters

bytes **byte[]** Contents of *string* represented as a byte[].

substring

Returns a substring of a given string.

Input Parameters

inString **String** String from which to extract a substring.

beginIndex **String** Beginning index of the substring to extract (inclusive).

endIndex **String** Ending index of the substring to extract (exclusive). If this parameter is null or empty, the substring will extend to the end of *inString*.

Output Parameters

value **String** Substring from *beginIndex* and extending to the character at *endIndex* - 1.

tokenize

Tokenizes a string using specified delimiter characters and generates a String List from the resulting tokens.

This service does not return delimiters as tokens.

Input Parameters

inString **String** String you want to tokenize, that is, break into delimited chunks.

delim **String** Delimiter characters. If null or empty, the service uses the default delimiters `\t\n\r`, where t, n, and r represent the white space characters tab, new line, and carriage return.

Output Parameters

valueList **String List** Strings containing the tokens extracted from *inString*.

toLower

Converts all characters in a given string to lowercase.

Input Parameters

inString **String** String to convert.

language **String** Optional. Lowercase, two-letter ISO-639 code. If this parameter is null, the system default is used.

country **String** Optional. Uppercase, two-letter ISO-3166 code. If this parameter is null, the system default is used.

variant **String** Optional. Vendor and browser-specific code. If null, this parameter is ignored.

Output Parameters

value **String** Contents of *inString*, with all uppercase characters converted to lowercase.

toUpper

Converts all characters in a given string to uppercase.

Input Parameters

inString **String** String to convert.

language **String** Optional. Lowercase, two-letter ISO-639 code. If this parameter is null, the system default is used.

country **String** Optional. Uppercase, two-letter ISO-3166 code. If this parameter is null, the system default is used.

variant **String** Optional. Vendor and browser-specific code. If null, this parameter is ignored.

Output Parameters

value **String** Contents of *inString*, with all lowercase characters converted to uppercase.

trim

Trims leading and trailing white space from a given string.

Input Parameters

inString **String** String to trim.

Output Parameters

value **String** Contents of *inString* with white space trimmed from both ends.

URLDecode

Decodes a URL-encoded string.

Input Parameters

inString **String** URL-encoded string to decode.

Output Parameters

value **String** Result from decoding *inString*. If *inString* contained plus (+) signs, they will appear in *value* as spaces. If *inString* contained *%hex* encoded characters, they will appear in *value* as the appropriate native character.

URLEncode

URL-encodes a string.

Encodes characters the same way that data posted from a WWW form is encoded, that is, the `application/x-www-form-urlencoded` MIME type.

Input Parameters

inString **String** String to URL-encode.

Output Parameters

value **String** Result from URL-encoding *inString*. If *inString* contained non-alphanumeric characters (except `[-_.*@]`), they will appear in *value* as their URL-encoded equivalents (% followed by a two-digit hex code). If *inString* contained spaces, they will appear in *value* as plus (+) signs.

fuzzyMatch

A given string is not exactly matched against a set of strings. If the match is above *similarityThreshold*, it returns the *matchedValue*. If more than one string has not exactly matched, then the first matched string is returned.

Input Parameters

<i>inString</i>	String (Required) Text to be matched. Text should not be empty or null.
<i>matchData</i>	String [] (Required) Array of strings, which are used for matching. If the string array value is either empty or null, it is not used for matching.
<i>similarityThreshold</i>	String (Optional) If the inexact match score is above the given threshold, then service output contains the <i>matchedValue</i> parameter. Default value is 0.65. Valid values should be between 0.0 and 1.0. Value 0.0 represents no match and value 1.0 represents an exact match.
<i>algorithm</i>	String (Optional) The algorithm used for an inexact match. Default value is Levenshtein. Supported algorithms are Levenshtein and JaroWinkler.

Output Parameters

<i>matchedValue</i>	String (Optional) If the inexact match is above <i>similarityThreshold</i> , then the returned value contains the matched string.
<i>similarity</i>	String (Optional) If the inexact match is above <i>similarityThreshold</i> , then it contains a similarity score. It provides the measure of how close the match is. The returned value can be between 0.0 and 1.0. Value 0.0 represents no match and value 1.0 represents an exact match.

Usage Notes

For more information about the algorithms, see http://en.wikipedia.org/wiki/Levenshtein_distance and http://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance

Hashtable

The following hashtable elements are available:

Element	Description
<code>containsKey</code>	Checks for the existence of a hashtable element.
<code>createHashtable</code>	Creates a hashtable object.
<code>get</code>	Gets the value for a specified key in the hashtable.
<code>listKeys</code>	Lists all the keys stored in the hashtable.
<code>put</code>	Adds a key/value pair in the hashtable.
<code>remove</code>	Removes a key/value pair from the hashtable.
<code>size</code>	Gets the number of elements in the hashtable.

Summary of Hashtable elements

containsKey

Checks for the existence of a hashtable element.

Input Parameters

hashtable **java.util.Hashtable** Hashtable in which to check for the existence of a hashtable element.

key **String** Hashtable element to be checked for.

Output Parameters

containsKey **String** Indicates whether the specified hashtable element exists. A value of:

- `true` indicates that the element exists.

- `false` indicates that the element does not exist.

createHashtable

Creates a hashtable object.

Input Parameters

None.

Output Parameters

hashtable **java.util.Hashtable** The new hashtable object.

get

Gets the value for a specified key in the hashtable.

Input Parameters

hashtable **java.util.Hashtable** Hashtable from which to retrieve the specified value.

key **String** Key of the hashtable element whose value is to be retrieved.

Output Parameters

value **Object** Value of the input hashtable element.

listKeys

Lists all the keys stored in the hashtable.

Input Parameters

hashtable **java.util.Hashtable** Hashtable from which the keys are to be listed.

Output Parameters

keys **String[]** List of keys stored in the input hashtable.

put

Adds a key/value pair in the hashtable.

Input Parameters

hashtable **java.util.Hashtable** Hashtable to which the key/value pair is to be added.

key **String** Key of the element to be added to the hashtable.

value **Object** Value of the element to be inserted into the hashtable.

Output Parameters

hashtable **java.util.Hashtable** Hashtable object after the insertion of the key/value pair.

remove

Removes a key/value pair from the hashtable.

Input Parameters

hashtable **java.util.Hashtable** Hashtable from which to remove the key/value pair.

key **String** Key of the hashtable element to be removed.

value **Object** Value of the hashtable element to be removed.

Output Parameters

hashtable **java.util.Hashtable** Hashtable object after the key/value pair is removed.

value **Object** Value of the hashtable element that was removed. Returns `null` if the input *key* is not found in the hashtable.

size

Gets the number of elements in the hashtable.

Input Parameters

hashtable **java.util.Hashtable** Hashtable from which the number of elements stored in it is to be retrieved.

Output Parameters

size **String** Number of elements in the hashtable.

IO

You can use the IO elements to convert data between `byte[]`, characters, and `InputStream` representations. These services are used for reading and writing bytes, characters, and streamed data to the file system.

These services behave like the corresponding methods in the `java.io.InputStream` class. For more information about `InputStreams`, see the Java documentation.

Summary of IO elements

Note: These services can be invoked only by other services. Streams cannot be passed between clients and the server, so these services will not execute if they are invoked from a client.

Element	Description
bytesToStream	Converts a <code>byte[]</code> to <code>java.io.ByteArrayInputStream</code> .
streamToBytes	Creates a <code>byte[]</code> from data that is read from an <code>InputStream</code> .

bytesToStream

Converts a `byte[]` to `java.io.ByteArrayInputStream`.

Input Parameters

<i>bytes</i>	byte[] The byte array to convert.
<i>length</i>	String Optional. The maximum number of bytes to read and convert. If <i>length</i> is not specified, the default value for this parameter is the length of the input byte array.
<i>offset</i>	String Optional. The offset into the input byte array from which to start converting. If no value specified, the default value is zero.

Output Parameters

<i>stream</i>	java.io.ByteArrayInputStream An open <code>InputStream</code> created from the contents of the input <i>bytes</i> parameter.
---------------	---

Usage Notes

This service constructs *stream* from the byte array using the constructor `ByteArrayInputStream(byte[])`. This constructor does not make a copy of the byte array, so any changes to *bytes* will be reflected in the data read from the stream.

streamToBytes

Creates a `byte[]` from data that is read from an `InputStream`.

Input Parameters

<i>stream</i>	java.io.InputStream The <code>InputStream</code> that you want to convert.
---------------	---

Output Parameters

<i>bytes</i>	byte[] The bytes read from <i>stream</i> .
--------------	--

Usage Notes

This service reads all of the bytes from *stream* until the end of file is reached, and then it closes the `InputStream`.

Reference Data

Reference data is data that defines the set of permissible values to be used by other data fields. It is a collection of key-value pairs, which can be used to determine the value of a data field based on the value of another data field. For example, the value of a status field in an Application can be “Canceled” and that needs to be interpreted as “CN” in another Application.

Integration Cloud allows you to upload Reference Data from a text file containing tabular data separated by a character, for example, a comma, semicolon, and so on. You can access the uploaded Reference Data in Orchestrated Integrations as a list of documents, by using the Reference Data block and providing an appropriate name. You can filter the documents returned into the pipeline by the Reference Data block using the `findDocuments` service.

The Reference Data block appears under **Flow Control** in the Orchestrated Integration workspace, only after a Reference Data is created.

Note: You must have the required access privileges under **Settings > Access Profiles > Administrative Permissions > Functional Controls > Reference Data** if you want to create, update, or delete any Reference Data.

To add or edit a Reference Data

1. From the Integration Cloud navigation bar, click Integrations > Reference Data. The “My Reference Data” screen appears.
2. To edit an existing Reference Data, select a Reference Data from the “My Reference Data” screen and click Edit.
3. To create a new Reference Data, from the “My Reference Data” screen, click “Add New Reference Data”.
4. Provide a name and description of your Reference Data. Required fields are marked with an asterisk in the screen.
5. Click “Browse” and select a .txt or .csv file. Only a text file (.txt or .csv) having tabular data is supported. The maximum file size you can upload is 1 MB.
6. Click Next to define and preview the Reference Data. Select the field separator and the text qualifier.
7. Determine the encoding of the Reference Data file and from the “File Encoding” drop down list, select the same encoding. Click “Load Preview” to preview the data.

If you select an incorrect encoding, garbage characters will appear in the preview pane.

8. Click Next to review the Reference Data and then click Finish to create the Reference Data.

The new Reference Data appears in the “My Reference Data” screen.

Note: The **Reference Data** block will appear under **Flow Control** only after you have created a Reference Data and the Reference Data service will be available while creating an Orchestrated Integration. If a Reference Data is used by an Integration, you will not be able to delete the Reference Data.

Pipeline and Signatures

The pipeline is the general term used to refer to the data structure in which input and output values are maintained for an orchestration. It allows services in the flow to share data. The pipeline starts with the input to the orchestration and collects inputs and outputs from subsequent Applications and services in the orchestration. When an operation of an Application or a service in the orchestration executes, it has access to all data in the pipeline at that point.

Input and output parameters are the names and types of fields that the service requires as input and generates as output. These parameters are also collectively referred to as a signature.

The input side describes the initial contents of the pipeline. In other words, it specifies the variables that this orchestration expects to find in the pipeline at run time. The output side identifies the variables produced by the orchestration and returned to the pipeline.

Mapping adjusts the pipeline at any point in an orchestration. It makes pipeline modifications. You use mapping when you want to assign an initial set of input values in an orchestration. You do the mapping at the beginning of the orchestration and then use the Set Value modifier to assign values to the appropriate variables in Pipeline Output. You can insert transformers into the MAP step to perform the needed data transformations.

Integration Details

This screen allows you to view at which stage the Integration is running, the Applications and Accounts involved to create the Integration including the Look up Transformer, when the Integration was created or last modified, who created or last modified the Integration, when was the last execution, and whether the Integration is scheduled. You can also delete, edit, or expose the Integration as a REST service from this screen.

Option	Description
Develop	Click to view the My Integrations screen.
Execution Results	Click to view the Execution Results screen.
Created on	Displays the date when the Integration was created.
Created by	Displays the user who created the Integration.
Applications	Displays the Applications used to create the Integration including the Look up Transformer. Click an Application to view the specific Application details screen.
Stages	Displays different stages of the Integration development. You can pull Integrations from all other stages except from the Development stage.
Edit	Click to modify the Integration.
Delete	Click to delete the Integration from a stage.
Accounts	Displays the Accounts used by the Integration. Click an Account to view the specific Account configuration screen.
Last modified	When and by whom was the Integration last modified.
Status	If the Integration in a stage is scheduled, then the status of the Integration displays Scheduled , else it appears as Not Scheduled . The Status appears as Schedule Paused if the Integration is paused.
Last execution	When was the Integration last executed. A warning message appears if the last execution was not successful.
Resume	Click to start the Integration that was paused.
Schedule	Click to schedule the Integration to run immediately or at a specified date and time. You can also define a recurrence pattern to run the Integration.
Run Now	Click to submit the Integration for execution.

Option	Description
Pull	<p>Click to pull an Integration from a preceding stage into this stage. You can pull an Integration after you have configured the Accounts for each stage in the Account Configuration page.</p>
Expose as a REST service	<p>Check this option if you want to trigger the execution of an Integration from an external system. By default, Integration flows built in Integration Cloud are not accessible using REST. This feature provides you with one more option to trigger Integration executions from a software application, for example, a REST Client, apart from manual and scheduled integrations from the user interface.</p> <p>Once the flow is exposed as a REST service, the REST URL appears. Click the Show Advanced Options link to view the HTTP Method, sample JSON input, and the parameters which are required to invoke this Integration from an external system.</p> <p>Provide the usage URL, HTTP Method, modified or the sample JSON input, and necessary parameters in the external program, including the required security credentials (user name and password) while submitting the REST service.</p> <p>After the request is sent, the response will contain a status indicating whether the Integration has been submitted for execution. The response will also contain a reference to the execution result identifier so that a new REST call can be made later to get the execution results.</p> <p><i>Application Status Codes</i></p> <ul style="list-style-type: none"> ■ 0 - SUCCESS: Successfully submitted the Integration for execution. ■ -1 - ERROR: Problem while submitting the Integration for execution. <p><i>HTTP Status Codes</i></p> <ul style="list-style-type: none"> ■ 200 - OK ■ 201 - Created ■ 500 - Internal Server Error ■ 401 - Unauthorized User Error <p>To get the execution results, construct the URL of the new REST call from the URI field available in the <i>Response</i> section.</p> <p>To construct the URL of the new REST call, add the response URI obtained from <i>resultReference</i> in the <i>Response</i> section to:</p>

Option	Description
	<pre>https://<sub-domain>.webMethodscloud.com/integration</pre>
	<p>Example:</p> <pre>https://<sub-domain>.webmethodscloud.com/integration/rest/assembly/external/execution/result?resultReference=76fb5733-6a21-4b02-864f-5e958f698373</pre>
	<p><i>Application Status Codes</i></p>
	<ul style="list-style-type: none"> ■ 0 - SUCCESS ■ -1 - ERROR
	<p><i>HTTP Status Codes</i></p>
	<ul style="list-style-type: none"> ■ 200 - OK ■ 500 - Internal Server Error ■ 401 - Unauthorized User Error ■ 404 - Not Found
	<p>Note: You must provide your user name and password to execute the Integration from the external program, else you may encounter the 401 - Unauthorized User Error. Further, if the query response HTTP status code is 404 - Not Found, it means that either the Integration is not yet executed or the <i>resultReference</i> is incorrect.</p>

Execution Results

The **Execution Results** screen allows you to view the audit trail of all the executions that happened in a stage.

To view the execution results

1. From the Integration Cloud navigation bar, click Integrations. The My Integrations screen appears.
2. From the My Integrations screen, select the Integration for which you want to view the execution results.
3. Click the Execution Results link to view the Execution Results screen.

You can also click the Execution Results link in the Home page or click **Integrations > Execution Results** to view the **Execution Results** screen.

4. Select the **Integration**, the **Stage**, and the time period for which you want to view the execution results.

The **Integration** and the **Stage** fields will be automatically populated if you select the **Execution Results** link for an Integration from the **My Integrations** screen.

5. Click Show Results to view the execution results.