

Data Synchronizer: Keeping Your Apps and Mobile Devices Constantly in Sync

Table of Contents

page

Keeping Your Apps and Mobile Devices Constantly in Sync	1
Architecture Overview	2
Flat File Connector	8
Connector Development	9
Data Synchronizer Requirements	10
Stay Connected and in Sync	10

Keeping Your Apps and Mobile Devices Constantly in Sync

The inability for business critical applications, collaboration systems and mobile devices to interact with each other in real time creates considerable frustration for users and the IT personnel responsible for supporting them. Individuals need critical data and information to be available exactly when and where they need it, but it can be difficult for IT to ensure that the required integration and synchronization occurs.

As a result, users waste significant time switching between different applications and devices to get to the business-critical information they need. Still, the lack of integration between mobile devices and applications often prevents users from getting the critical information they need, when they need it.

The real-time synchronization in OpenText™ Data Synchronizer addresses these problems, letting users update information once and then leverage it anywhere at any time. Data Synchronizer enables organizations to bi-directionally synchronize data between different applications such as OpenText™ GroupWise, Microsoft SharePoint, SugarCRM and salesforce.com. In addition, the product synchronizes data between these applications and a wide variety of mobile devices, such as devices that use the iPhone, Android, Symbian, Palm and Windows Mobile operating systems.

By keeping users' applications and mobile devices constantly in sync, Data Synchronizer ensures that users always have access to the data they need, allowing organizations to:

- **Save users time.** Eliminates the need to keep switching between applications or manually keep data up to date in multiple applications and mobile devices
- **Improve collaboration efforts and mobile productivity.** Automatically pushes updates to collaboration data to relevant applications and mobile devices so even mobile users can always access the most up-to-date and accurate information
- **Facilitate IT administration.** Provides a central path through which an organization's collaboration data can flow, synchronizing that data among key applications

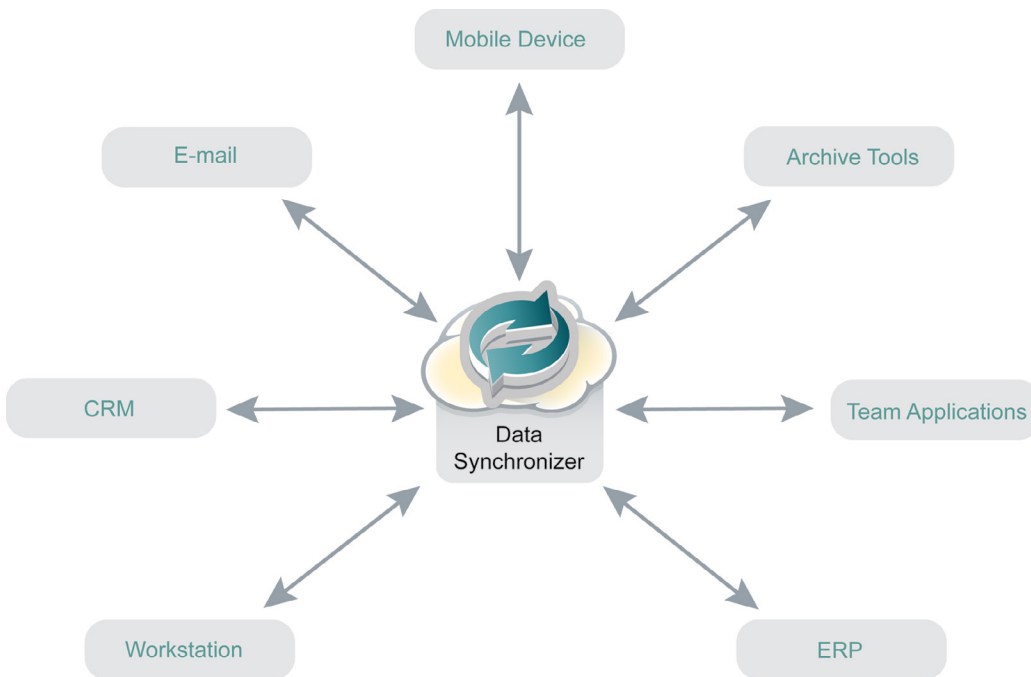
- **Enable managers to make informed decisions.** Allows for instant access and visibility into customer and sales data because it's synchronized in real time from applications and mobile devices into CRM or other management systems

Data Synchronizer enables organizations to bi-directionally synchronize data between different applications such as GroupWise, Microsoft SharePoint, SugarCRM and salesforce.com.

Architecture Overview

Data Synchronizer is a bi-directional, many-to-many synchronization engine that provides back-end synchronization of e-mail, calendar items, contacts and tasks to multiple systems. As a multi-source, multi-target synchronization engine, it's not tied to any specific application or device. This enables it to synchronize data and events among a broad array of systems, including traditional collaboration offerings as well as mission-critical enterprise applications such as CRM, ERP and other business solutions. Whenever changes occur in one of these connected systems, Data Synchronizer stores those changes in real time to allow other connected systems to access them.

Following a hub and spoke architectural model, Data Synchronizer has a synchronization engine that resides in the middle as the hub with different connectors acting as the spokes. It also employs channels that conduct the flow of data from the synchronization engine through the connectors to the connected systems.



Data Synchronizer synchronizes data and events among a broad array of systems, including traditional collaboration offerings as well as mission-critical enterprise applications such as CRM, ERP and other business solutions.

In this way, the design of Data Synchronizer has many similarities to that of NetIQ Identity Manager by OpenText™. While NetIQ Identity Manager synchronizes identities from different sources and targets, Data Synchronizer synchronizes data in a similar fashion.

In addition to the synchronization engine, connectors and channels, three other key components make up the Data Synchronizer architecture. These components include the Web administration service, Configuration Engine and Connector Manager.

Many-to-Many Synchronization Engine

The synchronization engine provides the common ground between the different applications, systems and devices connected to Data Synchronizer. As users update information in their connected applications or devices, the synchronization engine stores those changes in real time to allow other connected applications and devices to automatically access that data. In essence, it acts as the hub for the flow of data that passes among the different application-specific connectors and the mobility connector.

Since it's a many-to-many synchronization engine, organizations don't have to rely on a single specific application for synchronization to other applications. The synchronization engine synchronizes from multiple applications to multiple targets. This means users can log into any one of their connected applications or mobile devices, knowing that their data will be synchronized. As a result, users just need to update information once to be able to leverage it anywhere at any time.

As e-mail, calendar, contact, event and other collaboration data changes in different business and collaboration systems, Data Synchronizer seamlessly synchronizes those changes to all other relevant systems—in real time. This seamless real-time integration between enterprise systems, third-party applications and mobile devices eliminates the need to manually add the same information into multiple systems. Users enter data once, and it's automatically pushed to other relevant applications and mobile devices so they can access their data from anywhere, anytime.

Additionally, since it's a server-side synchronization engine, the solution is not bound to any particular client or application. Nor does it necessarily require any changes to an application in order to operate. As a result, users can use their tool of choice and know that they will be synchronized and kept up to date.

SCALABLE ARCHITECTURE

While a single Data Synchronizer synchronization engine can support multiple connectors, the solution has been designed with a scalable architecture that can support the deployment of multiple synchronization engines supporting multiple connectors. Even though the initial release only officially supports a single synchronization engine implementation, it has been designed to allow multi-engine implementation. With this in mind, it's expected that such implementations will be supported in the future.

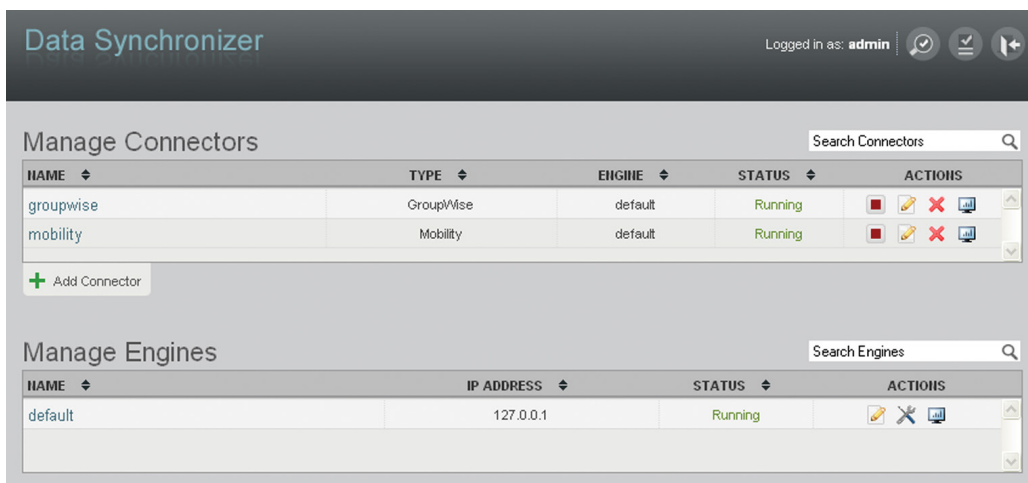
Data Synchronizer Connectors

We provide Data Synchronizer connectors for GroupWise, SharePoint, SugarCRM and salesforce.com, as well as a flat file connector. We also provide a mobility connector that synchronizes e-mail, contact and calendar data for most mobile devices that use the iPhone (2.0, 2.0. 3.1 or later), Windows Mobile (6.0, 6.1 and 6.5), Palm, Symbian (Series 60 third, fourth or fifth edition) and Android (2.0) operating systems. For example, when the mobility connector and GroupWise connector are installed, GroupWise content can be synchronized to an iPhone or Android mobile device. Likewise, if the SugarCRM connector is added, GroupWise contact information can also be synchronized with the contact information in SugarCRM.

In the future, we also plan to make connectors available for OpenText™ Vibe, Documentum, SAP, Exchange and others. In addition to these connectors, enterprises and third-party developers can customize or create their own connectors using the Data Synchronizer SDK.

Data Synchronizer helps organizations to:

- Save users time
- Improve collaboration efforts and mobile productivity
- Facilitate IT administration
- Enable managers to make informed decisions



Channels

As mentioned previously, Data Synchronizer employs concepts that are similar in nature to those of NetIQ Identity Manager. One of these concepts deals with the idea of a two channel system with a source channel for outgoing events and a SINK channel for incoming events. In Data Synchronizer these channels exist in both the synchronization engine and in the individual connectors.

The source channel in the synchronization engine is responsible for taking item or event data from its cache and passing it to the connector. After the connector translates the data into the appropriate application specific formation, the connector's SINK channel will pass that information to the application. For data coming from an application to the engine, the connector's source channel is responsible for taking data from the application, while the engine's SINK channel will place the application-neutral formatted data into its cache.

TRANSLATION AND FILTERS

Data Synchronizer connectors provide the integration between the different applications, enterprise systems and devices connected to Data Synchronizer. The connectors transfer data between two or more applications, with each connector acting as a code converter and data filter. This data flows through the Data Synchronizer system in the form of XML files that can be processed by the connectors and the synchronization engine. As an XML file moves from an application-specific connector to the synchronization engine, the connector filter translates the XML file from an application-specific format to an application-neutral format that is stored in the synchronization engine database. In its application-neutral format, the XML file can then be consumed by different application-specific connectors, which can translate it into the application-specific format required by their associated applications.

In most cases, the application-specific format used by any connector will be based on the default or standard set of APIs used by that specific application. For example, the SharePoint connector uses the SharePoint Web services API to access data and push data through the system. Similarly, the GroupWise connector utilizes the SOAP interface to synchronize e-mail, tasks, calendar data and contact information. The use of standard or native APIs allows each application to be built specifically for its target application.

SYNCHRONIZATION FILTERS

Additionally, depending on the needs and purpose of the different applications, each application will have different types of data that it will allow to be synchronized. Some of the common data types (referred to as data items), include e-mail messages, appointments, tasks, notes, address books, contacts and folders. Not all connectors will necessarily support all of the same types of data items. Data item synchronization will vary based on application functionality.

Data Synchronizer also synchronizes changes that occur to data items within a connected application, system or device. These changes (known as events) might be represented by actions such as an item addition, modification, move or deletion.

As data items and events pass through a connector, various filters applied to the data determine how the data is synchronized. Depending on the connector design, administrators and users can often use filters to determine which data items and events they want synchronized.

Not only can filters filter out certain content or fields, but they can transform one data type to another, such as turning task data into e-mail data. Filters can also look for certain triggers that affect how the filter behaves.

Filters

Filter Name	Type	Actions
applicationSource	engine.filter.xstt	
schemamap	engine.filter.xstt	
eventFilter	engine.filter.xstt	
usermapfilter	engine.filter.targetFromAI	
folderTracker	engine.filter.folderTracker	

Filter Name	Type	Actions
engineSource	engine.filter.xstt	
eventFilter	engine.filter.xstt	
usermapfilter	engine.filter.targetToAI	
folderTracker	engine.filter.folderTracker	
schemamap	engine.filter.xstt	

For example, a trigger might cause a filter to treat a data item differently based on where that data was originally stored on the source application.

A connector's filters can be configured through the Data Synchronizer Web Admin console. For example, for the GroupWise connector, an administrator can choose from a list of different events (add events, modify events, move events and delete events) that it wants to synchronize.

Add Events To Sync	<ul style="list-style-type: none">AddressBookAddAddressBookItemAddFolderAddFolderItemAddProxyAccessAddPersonalGroupItemAdd
Modify Events To Sync	<ul style="list-style-type: none">AddressBookModifyAddressBookItemModifyItemAcceptItemDeclineItemUnacceptItemArchive
Move Events To Sync	<ul style="list-style-type: none">FolderItemMove
Delete Events To Sync	<ul style="list-style-type: none">AddressBookDeleteAddressBookItemDeleteFolderDeleteFolderItemDeleteItemDeleteItemUndelete

One of the standard filters in Data Synchronizer provides a circular sync check. As data passes from the connector to the synchronization engine, this filter makes sure that it's not synchronizing the same data again in a circular fashion. This circular check filter prevents the creation of duplicate information.

The default filters that come with a connector are typically sufficient to enable the translation of data from an application-specific format into an application-neutral format and vice versa, as well as apply other basic filter operations. However, since these filters are made up of XSLT files, administrators familiar with XSLT also have the ability to manually configure them if necessary.

FILTER FUNCTIONS

While an administrator can define additional filter functions using the connector's custom namespace, the following are standard provided functions in the Data Synchronizer namespace:

- `def logMessage(self, context, message, level='warning')`: Logs a message
- `def getSettingsForTarget(self, context, targetDn)`: Returns a user's settings
- `def getConnectorSettings(self, context)`: Returns a connector's settings
- `def getEngineSettings(self, context)`: Returns the sync engine's settings
- `def getConfigEngineSettings(self, context)`: Returns the configuration engine's settings
- `def getMatchingUserDN(self, context, sourceName)`: Returns the dn of a user with a unique cn provided
- `def dnExists(self, context, dn)`: Checks if the dn is a valid dn in the tree
- `def isTargetEnabled(self, context, dn)`: Checks if the user is in the user table for the connector
- `def base64Encode(self, context, message)`:
- `def base64Decode(self, context, encodedMessage, returnAsNodeSet=False)`:
- `def stripHTML(self, context, text, unescapeFirst=False)`: Removes html from the text
- `def itemIDToObjectID(self, itemID)`:
- `def objectIDToItemID(self, objectID)`:

Other Data Synchronizer Components

In addition to its synchronization engine and connectors, the following three components also play a critical role in the operation of the Data Synchronizer system:

- **Web Administration Service** provides a Web-based interface called Synchronizer Web Admin to facilitate administration and management of the Data Synchronizer system. It allows administrators to add and remove connectors, as well add or remove users for specific connectors and configure their individual synchronization settings. Users can also use the Synchronizer Web Admin to configure and control their personal synchronization settings for each of their connected applications and devices.
- **Synchronizer Configuration Engine** provides communication between Synchronizer Web Admin and the synchronization engine. The configuration engine passes configuration information from the synchronization engine to Synchronizer Web Admin for viewing. It also passes configuration changes back to the synchronization engine for implementation.
- **Connector Manager** provides communication between the synchronization engine and connectors.

LDAP INTEGRATION

While not required, Data Synchronizer can leverage an LDAP directory to further facilitate administration of user and group management.

Flat File Connector

In addition to the mobility connector and various application connectors, Data Synchronizer also includes a flat file connector. This connector can take item and event data cached in the synchronization engine and push it out in a human readable XML format to a specified directory or folder on the network. Once placed in the specified folder, it can be consumed by other applications or manually accessed by users.

The flat file connector also has the ability to periodically poll a specified folder for newly stored data. When it identifies new data, it can grab that data and pass it back to the synchronization engine for consumption by connected applications or devices.

While it operates on a fairly simple and straightforward concept, its inherent flexibility and open nature make the flat file connector quite powerful. It presents the opportunity for organizations to extend the capabilities of Data Synchronizer in a variety of ways, limited only by imagination.

Flat File Connector Use Cases

The flat file connector, in essence, provides a semi-technical programming interface that allows organizations to customize and extend the capabilities of Data Synchronizer. Leveraging the connector, an internal programmer could create a custom application that when a certain event happens, it generates certain information in an XML file format and then stores it in a specified folder. Once in the folder, the flat

The flat file connector allows organizations to extend the capabilities of Data Synchronizer in a variety of ways, limited only by imagination.

Whether in the office,
at home or on the road,
Data Synchronizer
ensures that users
always have access to
the right information,
in the right place,
at the right time.

file connector can grab it, transform it and store it in the synchronization engine where it can be used by other connected applications.

One possible use would be to utilize the flat file connector to retrieve and forward reports generated by a certain application. The application could generate the report as an XML file and store it in a specified folder. As the flat file connector polls that folder, it discovers the new report, transforms it into an application-neutral format and stores it in the synchronization engine database where it can be consumed by other connected applications or sent to a specific user.

The flat file connector could also be used to pre-populate a new user's GroupWise, SharePoint or SugarCRM system with standard information. To accomplish this, a file can be manually created in the proper XML format with the desired information and then stored in a specified directory. The flat file connector can then grab, transform and store that data in the synchronization engine cache. When a new account is created for a user, its application specific connector can then automatically pull that data from the synchronization engine and pre-populate the user's account accordingly.

Another use case might be utilizing the flat file connector to provide basic archiving in a standard text format. For example, an organization might have a policy that certain data needs to be archived for some extended period such as 50 years. The organization will want to use a standard format such as XML to store that data to ensure that it will still be readable after that extended period of time. As data that meets the criteria to be archived enters the synchronization engine cache, the flat file connector can grab it and store it in the desired XML format in a specified archival storage location.

Connector Development

It's important to note that connectors won't be limited to what we provide. By leveraging the Data Synchronizer software development kit and its open API, organizations and partners will be able to create custom connectors for their own systems as well as third-party solutions. The open nature of Data Synchronizer creates limitless opportunities for organizations to create highly productive environments where users can share relevant content from a wide array of collaboration and enterprise systems, such as CRM, ECM and document management systems; cloud-based social applications; and even solutions that compete with our collaboration offerings.

To further assist in connector development, we provide a generic SOAP connector, which accesses Data Synchronizer using standard SOAP calls. This connector is primarily intended to be used as an example connector, providing a guide or reference point for someone who wants to develop a connector. They can use the connector as a model to allow them to see how the synchronization engine transforms data coming out of the GroupWise system.

Data Synchronizer Requirements

Currently, Data Synchronizer must be installed as an add-on to a SUSE Linux Enterprise Server 11 64-bit server. Data Synchronizer customers will automatically receive SUSE Linux Enterprise Server entitlements for as many servers as are required to run Data Synchronizer. In the future, we plan to expand platform support for Data Synchronizer to include Windows. Additionally, the Data Synchronizer installation process will create a PostgreSQL database that will be used for the caching of the data items and events to be synchronized. Other databases will be supported in the future.

Data Synchronizer comes with the Mobility Pack, which includes both the GroupWise connector and the mobility connector. When you install Data Synchronizer the complete Mobility Pack will be automatically installed. To help prepare for the installation, our documentation provides a mobility pack installation summary sheet to assist in the gathering of details such as LDAP server information, LDAP credentials, LDAP containers, GroupWise post office agents and mobile device ports. Once Data Synchronizer has been installed and the Mobility Pack has been configured, organizations can install other available connectors as well.

To take advantage of the installed GroupWise connector for synchronizing e-mail, tasks, calendar data and contact information, organizations need to be running GroupWise 8.0.2. To use the mobility connector, an account will need to be set up on users' mobile devices to allow them to synchronize using the mobility connector. While the actual configuration will vary based on device type, the account will typically need to be configured with the user's GroupWise e-mail address, GroupWise mailbox password, e-mail account type, Data Synchronizer server IP address or hostname and a secure connection certificate. Once the mobile device connects to the Data Synchronizer system (depending on GroupWise administration settings), the user's GroupWise personal address books, past e-mail messages (three days or newer), past calendar items (14 days or newer), and all future e-mail and calendar items will be synchronized to the mobile device.

Stay Connected and in Sync

The goal of Data Synchronizer is to deliver the real-time synchronization that users need between diverse applications and mobile devices in order to update and access the data they need, when they need it and where they need it. Its bi-directional, many-to-many synchronization engine improves an organization's collaboration efforts seamlessly. It does this by synchronizing e-mail, calendar, contact, event and other collaboration data changes among different business and collaboration systems in real time. Since it's not tied to any specific collaboration solution, it enables the synchronization of data and events between traditional collaboration systems and mission-critical enterprise applications, including GroupWise, Microsoft SharePoint, SugarCRM, salesforce.com, mobile devices and more. Organizations can use or customize existing connectors or create new ones to address specific internal or customer needs.

Whether in the office, at home or on the road, Data Synchronizer ensures that users always have access to the right information, in the right place, at the right time.

Learn more at
www.microfocus.com/opentext

Connect with Us

[OpenText CEO Mark Barrenechea's blog](#)

