



PUBLIC

Document Version: 2608 – 2026-08

NDA Data Export API

SAP Ariba APIs

SAP Ariba Developer Portal

SAP Ariba Contracts



Content

NDA Data Export API.	3
Endpoints Included in the NDA Data Export API.	3
Workflow for NDA Data Export API.	4
Retrieving Metadata for a Document Type Using the NDA Data Export API.	4
View Templates.	6
Creating a View Template using the NDA Data Export API.	7
Updating a View Template Using the NDA Data Export API.	9
Deleting a View Template Using the NDA Data Export API.	11
Searching View Templates Using the NDA Data Export API.	12
Exporting Data Using the NDA Data Export API.	14
Response Codes Returned by NDA Data Export API.	17

NDA Data Export API

This guide is for developers intending to implement the NDA Data Export API to export transactional data related to non-disclosure agreements (NDA).

Developers can create a client application on the SAP Ariba developer portal to retrieve the metadata of a document type, filter the metadata to export using view templates, and export the filtered data.

This guide applies to:

- SAP Ariba APIs
- SAP Ariba developer portal
- SAP Ariba Contracts
- SAP Ariba Strategic Sourcing Suite

Prerequisites

- To create an application and request access to use the SAP Ariba APIs relevant to this feature, you must have access to the SAP Ariba Developer Portal. For details refer to [Steps to Start Using the SAP Ariba APIs](#) in [Help for the SAP Ariba Developer Portal](#).
- All queries made to this API must be authenticated using OAuth authentication. For details refer to [SAP Ariba Developer Portal Authentication](#) in [Help for the SAP Ariba Developer Portal](#).

Endpoints Included in the NDA Data Export API

The NDA Data Export API supports the following endpoints:

GET /ndas/metadata	Retrieves the metadata of the specified document type.
POST /ndas/viewTemplates	Creates a view template containing filter criteria for a specified document type.
PUT /ndas/viewTemplates/{viewName}	Updates the content of the specified view template.
DELETE /ndas/viewTemplates/{viewName}	Deletes the specified view template.
GET /ndas/viewTemplates/{viewName}	Fetches the details of the specified view template.
GET /ndas/viewTemplates	Fetches the details of all the available view templates.

GET /ndas/pages/{pageNumber}

Exports the filtered data of a specific document type.

GET /ndas/{id}/files

Downloads the exported data of a specific document type.

Workflow for NDA Data Export API

The workflow for the NDA Data Export API includes the following steps:

1. **Metadata Retrieval:** Administrators can retrieve the data for a specific document type.
2. **View Management:** Administrators can create view templates to define the criteria to filter the metadata to export. They can also modify and use an existing view template.
3. **Export Data:** Administrators can export the data for a specified document type and download the exported data in .docx or .pdf file formats.

Retrieving Metadata for a Document Type Using the NDA Data Export API

Use this procedure to implement the NDA Data Export API in a client application to retrieve the metadata for a document type.

Context

The `GET /ndas/metadata` endpoint of the NDA Data Export API enables you to retrieve the metadata of a specific document type. The valid document types are **NDAWorkspace**, **NDADocusignStatus**, and **NDATemplate**.

Note

In the sample URLs in this topic, replace `{{runtime_url}}` with the desired runtime URL from the **Environment Details** table on the SAP Ariba Developer Portal discovery page for this API.

Procedure

Invoke the `GET /ndas/metadata` endpoint.

Example

```
GET {{runtime_url}}/ndas/metadata?  
user=myUser&passwordAdapter=myPassword&realm=myRealm&documentType=NDATemplate
```

Where,

user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.
realm	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.
documentType	Specifies the name of the document type for which the metadata is to be retrieved. This is a mandatory value.

Results

If the request is successful, the API returns the metadata of the specified document type. The metadata consists of the document type properties along with attributes such as **title**, **description**, **type**, and **filter**. The value in the **filter** attribute defines whether a property can be used in the filter criteria of a view template.

Sample Code

```
"name": "NDATemplate",
"properties": [
  {
    "title": "timeCreated",
    "description": "Time Created",
    "filter": true,
    "type": "date"
  },
  {
    "title": "createdBy",
    "description": "Created By User",
    "filter": true,
    "type": "string"
  },
  {
    "title": "legalEntity",
    "description": "Template legal entity",
    "filter": false,
    "type": "string"
  },
  {
    "title": "modifiedBy",
    "description": "Last Modified by User",
    "filter": false,
    "type": "string"
  },
  {
    "title": "publishedDate",
    "description": "Template published date",
    "filter": false,
    "type": "date"
  },
  {
    "title": "timeUpdated",
    "description": "Last Time Updated",
    "filter": false,
    "type": "date"
  }
],
```

```

{
  "title": "title",
  "description": "NDA template title",
  "filter": false,
  "type": "string"
},
{
  "title": "documentUniqueName",
  "description": "Template unique name",
  "filter": false,
  "type": "string"
},
{
  "title": "id",
  "description": "Primary Id to identify record",
  "filter": false,
  "type": "string"
},
{
  "title": "status",
  "description": "Template status",
  "filter": false,
  "type": "string"
},
{
  "title": "uniqueName",
  "description": "UniqueName",
  "filter": false,
  "type": "string"
},
{
  "title": "active",
  "description": "Is Active",
  "filter": false,
  "type": "int"
}
]
}

```

For information about the response codes returned by the API, see [Response Codes Returned by NDA Data Export API \[page 17\]](#).

Related Information

[View Templates \[page 6\]](#)

View Templates

A view template is a set of filter criteria that can be used to filter the metadata of a document type.

Use view templates with the NDA Data Export API as an efficient alternative to the manual process of appending filter criteria to an endpoint. View templates help manage simple and complex filter criteria separately and can be provided as an input parameter for exporting data.

The NDA Data Export API enables you to manage view templates for different document types. You can:

- Create a view template for a document type
- Update a view template
- Delete a view template
- Search for a specific view template or view all the available view templates

[Creating a View Template using the NDA Data Export API \[page 7\]](#)

[Updating a View Template Using the NDA Data Export API \[page 9\]](#)

[Deleting a View Template Using the NDA Data Export API \[page 11\]](#)

[Searching View Templates Using the NDA Data Export API \[page 12\]](#)

Creating a View Template using the NDA Data Export API

Use this procedure to implement the NDA Data Export API in a client application to create a view template for a document type.

Context

After you've retrieved the metadata for a document type, select the fields to include in the criteria for the view template. The fields that you select must have the `filter` attribute set to `true`. For information about retrieving the metadata for a document type, see [Retrieving Metadata for a Document Type Using the NDA Data Export API \[page 4\]](#).

The `POST /ndas/viewTemplates` endpoint of the NDA Data Export API enables you to create a view template for a specific document type.

Note

In the sample URLs in this topic, replace `{{runtime_url}}` with the desired runtime URL from the **Environment Details** table on the SAP Ariba Developer Portal discovery page for this API.

Create a request body containing the required filter criteria. The variables you can use for constructing the query body are as follows:

Variable	Description
<code>viewName</code>	Specify the name of the view template. This is a mandatory value.
<code>documentType</code>	Specify the document type for which you are creating the view template. The valid document types are: • NDAWorkspace • NDADocuSignStatus • NDATemplate This is a mandatory value.

Variable	Description
<i>selectFields</i>	Specify the fields from the metadata of the document type that you want to filter. Use the comma-separated format to provide multiple values.
<i>filterExpressions</i>	Specify additional filter criteria. This variable has three attributes that form a set of filter expression. The attributes are: field: Specify the name of the field that has the filter attribute set to true. To identify the eligible fields, check the metadata retrieved for the document type. op: Specify the filter operator. The supported filter operators are: =, <>, >, <, >=, <= and IN. All operators except the IN support evaluation against a single value. The IN operator supports evaluation against multiple values (provided in a comma-separated format in the <i>value</i> attribute). The IN operator can be used with fields such as createdBy, recipientOrg, and signers. value: Specify the evaluation criterion. Note You can define a single filter expression or multiple expressions as required. When there are multiple filter expressions, the API fetches data that satisfies all the specified conditions.

Procedure

Invoke the POST `/ndas/viewTemplates` endpoint.

❖ Example

```
POST {{runtime_url}}/ndas/viewTemplates?
user=myUser&passwordAdapter=myPassword&realm=myRealm
```

Where,

user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.
realm	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

In the request body, specify the view name, document type, and the filter criteria that includes the fields from the metadata to be exported with the other filter expressions. A sample request body is as follows:

Sample Code

```
{
  "viewName": "TestNDATemplate",
  "documentType": "NDATemplate",
  "selectFields":
  ["title", "documentUniqueName", "status", "legalEntity", "publishedDate", "timeUpdated",
  "timeCreated", "createdBy", "modifiedBy", "id", "uniqueName", "active"],
  "filterExpressions": [
    {
      "field": "createdBy",
      "op": "=",
      "value": "adavis"
    },
    {
      "field": "timeCreated",
      "op": "<=",
      "value": "2022-11-14T00:00:00 IST"
    }
  ]
}
```

Results

If the request is successful, the API creates a view template for the specified document type.

For information about the response codes returned by the API, see [Response Codes Returned by NDA Data Export API \[page 17\]](#).

Updating a View Template Using the NDA Data Export API

Use this procedure to implement the NDA Data Export API in a client application to update a view template for a document type.

Context

The PUT `/ndas/viewTemplates/{viewName}` endpoint of the NDA Data Export API enables you to update a view template.

Note

In the sample URLs in this topic, replace `{{runtime_url}}` with the desired runtime URL from the **Environment Details** table on the SAP Ariba Developer Portal discovery page for this API.

Procedure

Invoke the PUT `/ndas/viewTemplates/{viewName}` endpoint.

❖ Example

```
PUT {{runtime_url}}/ndas/viewTemplates/myTemplate?
user=myUser&passwordAdapter=myPassword&realm=myRealm
```

Where,

{viewName}	Specifies the name of the view template that you want to update. This is a mandatory value.
user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.
realm	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

Create a request body with the required updates. A sample request body is as follows:

↗ Sample Code

```
{
  "viewName": "TestNDATemplate",
  "documentType": "NDATemplate",
  "selectFields":
  ["title", "documentUniqueName", "status", "legalEntity", "publishedDate", "timeUpdated",
  "timeCreated", "createdBy", "modifiedBy", "id", "uniqueName", "active"],
  "filterExpressions": [
    {
      "field": "createdBy",
      "op": "=",
      "value": "adavis"
    },
    {
      "field": "timeCreated",
      "op": "<=",
      "value": "2022-11-14T00:00:00 IST"
    }
  ]
}
```

Results

If the request is successful, the API updates the specified view template.

For information about the response codes returned by the API, see [Response Codes Returned by NDA Data Export API \[page 17\]](#).

Deleting a View Template Using the NDA Data Export API

Use this procedure to implement the NDA Data Export API in a client application to delete a view template for a document type.

Context

The **DELETE** `/ndas/viewTemplates/{viewName}` endpoint of the NDA Data Export API enables you to delete a view template of a document type.

Note

In the sample URLs in this topic, replace `{{runtime_url}}` with the desired runtime URL from the **Environment Details** table on the SAP Ariba Developer Portal discovery page for this API.

Procedure

Invoke the **DELETE** `/ndas/viewTemplates/{viewName}` endpoint.

Example

```
DELETE {{runtime_url}}/ndas/viewTemplates/myTemplate?
user=myUser&passwordAdapter=myPassword&realm=myRealm
```

Where,

{viewName}	Specifies the name of the view template that you want to delete. This is a mandatory value.
user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.
realm	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

Results

If the request is successful, the API deletes the specified view template.

For information about the response codes returned by the API, see [Response Codes Returned by NDA Data Export API \[page 17\]](#).

Searching View Templates Using the NDA Data Export API

Use this procedure to implement the NDA Data Export API in a client application to search for view templates.

Context

You can choose to search for a specific view template or view all the available view templates. The `GET /ndas/viewTemplates/{viewName}` endpoint of the NDA Data Export API enables you to search for a specific view template, whereas the `GET /ndas/viewTemplates` endpoint enables you to view all the view templates.

Note

In the sample URLs in this topic, replace `{{runtime_url}}` with the desired runtime URL from the **Environment Details** table on the SAP Ariba Developer Portal discovery page for this API.

Procedure

Choose whether you want to search for a specific view template or view all the available view templates.

- To search for a specific view template, invoke the `GET /ndas/viewTemplates/{viewName}` endpoint.

Example

```
GET {{runtime_url}}/ndas/viewTemplates/myTemplate?
user=myUser&passwordAdapter=myPassword&realm=myRealm
```

Where,

{viewName}	Specifies the name of the view template that you want to update. This is a mandatory value.
user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.
realm	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

If the request is successful, the API fetches the details of the specified view template. A sample code output is as follows:

Sample Code

```
{
  "viewName": "MyViewTemplate1",
  "viewInfo": {
    "documentType": "NDATemplate",
    "filterExpressions": [
```

```

        {
            "field": "createdBy",
            "op": "=",
            "value": "adavis"
        },
        {
            "field": "timeCreated",
            "op": "<=",
            "value": "2022-11-14T00:00:00 GMT"
        }
    ],
    "selectFields": [
        "title",
        "documentUniqueName",
        "status",
        "legalEntity",
        "publishedDate",
        "timeUpdated",
        "timeCreated",
        "createdBy",
        "modifiedBy",
        "rootId",
        "uniqueName",
        "active"
    ]
}

```

- To search for all the available view templates, invoke the GET `/ndas/viewTemplates` endpoint.

❖ Example

```

GET {{runtime_url}}/ndas/viewTemplates?
user=myUser&passwordAdapter=myPassword&realm=myRealm

```

Where,

user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.
realm	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

If the request is successful, the API fetches the details of all the available view templates. A sample code output is as follows:

📄 Sample Code

```

{
    "views": [
        {
            "viewName": "MyViewTemplate1",
            "viewInfo": {
                "documentType": "NDAWorkspace",
                "filterExpressions": [
                    {
                        "field": "signers",
                        "op": "=",
                        "value": "adavis"
                    }
                ]
            }
        }
    ],
}

```

```

        "selectFields": []
      },
    },
    {
      "viewName": "MyViewTemplate2",
      "viewInfo": {
        "documentType": "NDATemplate",
        "filterExpressions": [
          {
            "field": "createdBy",
            "op": "=",
            "value": "adavis"
          },
          {
            "field": "timeCreated",
            "op": "<=",
            "value": "2022-11-14T00:00:00 GMT"
          }
        ]
      },
      "selectFields": [
        "title",
        "documentUniqueName",
        "status",
        "legalEntity",
        "publishedDate",
        "timeUpdated",
        "timeCreated",
        "createdBy",
        "modifiedBy",
        "rootId",
        "uniqueName",
        "active"
      ]
    }
  ],
  ...
}

```

For information about the response codes returned by the API, see [Response Codes Returned by NDA Data Export API \[page 17\]](#).

Exporting Data Using the NDA Data Export API

Use this procedure to implement the NDA Data Export API in a client application to export data for a document type and download the exported data to a file.

Context

The GET `/ndas/pages/{pageNumber}` endpoint of the NDA Data Export API enables you to export the transactional data for a document type. The data consists of the NDA document records that are created using the specified document type. If the query returns a long list of search results, the data is paginated and you can view the search results on a particular page by specifying the page number.

You can also download the exported data in the **.docx** or **.pdf** file format using the GET `/ndas/{id}/files` endpoint of the NDA Data Export API.

📌 Note

In the sample URLs in this topic, replace `{{runtime_url}}` with the desired runtime URL from the **Environment Details** table on the SAP Ariba Developer Portal discovery page for this API.

Procedure

1. To export the data of a document type, invoke the GET `/ndas/pages/{pageNumber}` endpoint.

🔗 Example

```
GET {{runtime_url}}/ndas/pages/1?
user=myUser&passwordAdapter=myPassword&realm=myRealm&viewName=TestNDATemplate001
```

Where,

{pageNumber}	Specifies the value to display the search results on a particular page. The default value is 1.
user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated.
realm	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

You can also use the `maxpagesize` parameter in the header to specify the number of records to be displayed in each page. The default value is 100.

🔗 Example

You invoke the GET `{{runtime_url}}/ndas/pages/1?user=myUser&passwordAdapter=myPassword&realm=myRealm&viewName=TestNDATemplate001` endpoint.

Case 1:

Precondition:

- `maxpagesize = 100`
- `{pageNumber} = 1`
- Number of matching records: 50

All the records are displayed in the search results as the number of matching records is less than the value of the `maxpagesize` parameter.

Case 2:

Precondition:

- `maxpagesize = 25`
- `{pageNumber} = 1`

- Number of matching records: 50

Only the first 25 records are displayed in the search results as the number of matching records is more than the value of the `maxpagesize` parameter. To see the remaining search results, you must invoke the `GET /ndas/pages/{pageNumber}` endpoint again with the `{pageNumber}` value set to **2**.

→ Recommendation

We recommend that you keep the default value of the `maxpagesize` unchanged and invoke the `GET /ndas/pages/{pageNumber}` endpoint by incrementing the `{pageNumber}` value until there are no more search results and the API returns the 204 response code.

If the request is successful, the API returns a list of NDA document records that are created using the document type specified in the view template. To download the details of a specific document, note down the value of its `id` attribute. A sample code output is as follows:

Sample Code

```
{
  "count": 1,
  "nda": [
    {
      "title": "TestTemplateNDA API",
      "documentUniqueName": "NDATEMPLTDOC201",
      "status": "PUBLISHED",
      "legalEntity": "",
      "publishedDate": "2022-11-09T10:48:39 GMT",
      "timeUpdated": "2022-11-09T10:48:39 GMT",
      "timeCreated": "2022-11-09T10:48:39 GMT",
      "createdBy": "adavis",
      "modifiedBy": "adavis",
      "id": "6:NDATEMPLT:201",
      "uniqueName": "NDATEMPLT201",
      "active": 0
    }
  ]
}
```

2. To download the details of a specific NDA document, invoke the `GET /ndas/{id}/files` endpoint.

Example

```
GET {{runtime_url}}/ndas/6:NDATEMPLT:1/files?
documentType=NDATemplate&user=myUser&passwordAdapter=myPassword&realm=myRealm
```

Where,

{id}	Specifies the unique identifier of the NDA document record whose details you want to download. This is a mandatory value.
documentType	Specifies the document type of the NDA document record. This is a mandatory value.
user	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
passwordAdapter	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.

realm Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

If the request is successful, the API returns the details of the specified NDA document in a binary format. You can choose to download a copy of the content in the **.docx** or **.pdf** file format.

For information about the response codes returned by the API, see [Response Codes Returned by NDA Data Export API \[page 17\]](#).

Response Codes Returned by NDA Data Export API

The NDA Data Export API returns response codes when you submit a request. The following table describes the response codes:

Response Code	Description
200	Success.
204	No content or no data to export.
400	Bad Request. One or more parameters specified in the request are incorrect.
403	Forbidden. User does not have the required permissions to access the content.
404	No matching entries found.
413	Payload too large.
500	Internal server error.
501	Requested operation not supported. Feature not enabled.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

Copyright © 2026 Ariba, Inc. All rights reserved.

This documentation, as well as the Ariba solutions, software and/or services described in it, contain proprietary information. They are provided under a license or other agreement containing restrictions on use and disclosure and are also protected by copyright, patent and/or other intellectual property laws. Except as permitted by such agreement, no part of the document may be reproduced or transmitted in any form by any means, electronic, mechanical or otherwise, without the prior written permission of Ariba, Inc.

Ariba, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the documentation. The information contained in the documentation is subject to change without notice.

Ariba and Ariba products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Ariba, Inc. in the United States and other countries. Please see <http://www.ariba.com/legal/trademarks> for additional trademark information and notices.

Ariba Sourcing solutions (On Demand and software) are protected by one or more of the following patents, including without limitation: U.S. Patent Nos. 6,199,050; 6,216,114; 6,223,167; 6,230,146; 6,230,147; 6,285,989; 6,408,283; 6,499,018; 6,564,192; 6,871,191; 6,952,682; 7,010,511; 7,072,061; 7,130,815; 7,146,331; 7,152,043; 7,225,152; 7,277,878; 7,249,085; 7,283,979; 7,283,980; 7,296,001; 7,346,574; 7,383,206; 7,395,238; 7,401,035; 7,407,035; 7,444,299; 7,483,852; 7,499,876; 7,536,362; 7,558,746; 7,558,752; 7,571,137; 7,599,878; 7,634,439; 7,657,461; 7,693,747; 8,364,577; and 8,392,317. Patents pending.

Other Ariba product solutions are protected by one or more of the following patents:

U.S. Patent Nos. 6,199,050, 6,216,114, 6,223,167, 6,230,146, 6,230,147, 6,285,989, 6,408,283, 6,499,018, 6,564,192, 6,584,451, 6,606,603, 6,714,939, 6,871,191, 6,952,682, 7,010,511, 7,047,318, 7,072,061, 7,084,998; 7,117,165; 7,225,145; 7,324,936; 7,536,362; 8,364,577; and 8,392,317. Patents pending.

Certain Ariba products may include third party software or other intellectual property licensed from a third party. For information regarding software or other intellectual property licensed from a third party, go to <http://www.ariba.com/copyrights.cfm>.

