

# NISS WebSwap: A Web Service for Data Swapping

Ashish Sanil, Shanti Gomatam, Alan F. Karr

National Institute of Statistical Sciences, Research Triangle Park, NC 27709-4006, USA

{ashish,sgomatam,karr}@niss.org

## Abstract

Data swapping is a statistical disclosure limitation practice that alters records in the data to be released by switching values of attributes across pairs of records in a fraction of the original data.

Web Services are an exciting new form of distributed computing that allow users to invoke remote applications nearly transparently. National Institute of Statistical Sciences (NISS) has recently started hosting NISS Web Services as a service and example to the statistical sciences community. In this paper we describe and provide usage information for NISS WebSwap the initial NISS Web Service, which swaps one or more attributes (fields) between user-specified records in a microdata file, uploading the original data file from the user’s computer and downloading the file containing the swapped records.

## 1 Background

Several agencies, including federal and state statistical agencies collect individually identifiable data, and either release microdata, or produce statistical summaries for the public. Various statutes, regulations and policies require such agencies to protect individually identifiable data. Statistical disclosure limitation (SDL) methods [3, 5] refer to the techniques used to protect statistical tables and microdata.

Data swapping [2, 8, 9] is a strategy for protecting confidentiality in released microdata records. In the simplest case of its use, values of a single attribute (or variable) are swapped between randomly selected pairs of records, as illustrated in Table 1 for data extracted from the Current Population Survey (CPS) [6]. In this example, there are 8 categorical attributes — Age, Employer Type, Education Attained, Marital Status, Race, Sex, Average Weekly Hours Worked and Annual Salary. The *swapped attribute* is “Age.”

The purpose of data swapping is to introduce uncertainty into the mind of any data user or intruder as to whether records correspond to real data elements. In the example in Table 1, only two of the six records in the post-swap database are “real.” Swapping does not change the marginal distribution of any attribute. Nor does it change either the joint distribution of the unswapped attributes or — if more than one attribute is swapped<sup>1</sup> — the joint distribution of the swapped attributes. It may, however, distort joint distributions involving both swapped and unswapped attributes.

Conceptualize the microdata or data file as a matrix, with the rows representing individuals or observations (also called records), and columns representing attributes for which information on the observations is collected. Let  $n$  be the number of records, and let  $v$  be the total number of attributes in the microdata. Denote the  $j^{\text{th}}$  attribute in the microdata by  $X_j$ , and the  $i^{\text{th}}$  record by  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iv})$ . If we are swapping the values of attribute  $X_1$  for the pair  $(i, j)$ , the post-swap values of the  $i^{\text{th}}$  and  $j^{\text{th}}$  records would be  $(x_{j1}, x_{i2}, \dots, x_{iv})$  and  $(x_{i1}, x_{j2}, \dots, x_{jv})$ , respectively. Clearly not all attributes in a record will be swapped. We will call the subset of attributes that will be swapped the *swapped attributes* or *swapping attributes*. The fraction of the total  $n$  records in the microdata that are initially marked to be swapped will be called the *swap rate*, and is denoted by  $r$ .

---

<sup>1</sup>For our purposes, multiple attributes are always swapped simultaneously, not sequentially.

Record	Age	EmplType	Educ	MarStat	Race	Sex	AveHours	Salary
1	<25	Gov	HS	Marr	W	M	40	<\$50K
2	25–55	SE	Bach	Marr	NW	M	>40	<\$50K
3	25–55	Gov	Bach+	Unmarr	NW	F	>40	>\$50K
4	>55	Priv	Bach	Unmarr	W	F	>40	<\$50K
5	<25	Other	SomeColl	Marr	W	M	40	>\$50K
6	>55	Priv	Bach+	Marr	NW	F	40	>\$50K

Record	Age	EmplType	Educ	MarStat	Race	Sex	AveHours	Salary
1	<u>&gt;55</u>	Gov	HS	Marr	W	M	40	<\$50K
2	25–55	SE	Bach	Marr	NW	M	>40	<\$50K
3	<u>&lt;25</u>	Gov	Bach+	Unmarr	NW	F	>40	>\$50K
4	>55	Priv	Bach	Unmarr	W	F	>40	<\$50K
5	<u>25-55</u>	Other	SomeColl	Marr	W	M	40	>\$50K
6	<u>&lt;25</u>	Priv	Bach+	Marr	NW	F	40	>\$50K

Table 1: Example of Data Swapping, using data from the CPS. *Top*: Data prior to swapping. *Bottom*: Data following swapping, with “Age” swapped between records 1 and 6 and records 3 and 5. (For emphasis, swapped attributes are underlined.) Records 2 and 4 are not altered.

Swaps may not always result in different values for the swapped attributes. If we swap  $X_1$  (Age) for observations 2 and 3, both records remain unchanged; whereas if we swap  $X_1$  for observations 1 and 3, the pre- and post-swap records for these two observations will be different. We will call the second kind of swap a *true swap*, i.e., a true swap is a swap that results in different pre- and post-swap values for the records being swapped.

In some situations there may be conditions on pairs of records, defined by attributes not in the collection of swapping attributes, in order for one record in the pair to be a feasible swap candidate for the other. For instance, in the microdata array presented in Table 1, we may allow swaps of Age only between those records with the same value of Employer Type, so the pair (1,3) is a feasible swap, whereas the pair (1,2) is not. Alternatively, we may allow swaps of Age only between those records that have different values of Education, in this case the pair (3,5) is a feasible swap whereas the pair (3,6) is not. Such attributes whose values define the feasibility of swap candidates are called *constraining attributes*.

Any data release has an associated risk of disclosure, i.e., of data subject reidentification, and an associated utility value to users of the data. (Examples of such risk and utility measures are described in Section 6.) When swapping is used for purposes of SDL, several parameter choices are necessary: the swapped attributes, constraints and swapping rate. We conceptualize selection of these parameters as taking place in a risk-utility framework [4]. In virtually all cases, there is a tradeoff — higher utility implies higher risk. NISS WebSwap is usable, for example, to generate swapped databases under several parameter choices, and to compute risk-utility (R-U) frontiers that quantify the tradeoffs and inform the parameter selection process.

Sections 2, 3, 4 and 5 describe NISS WebSwap. Section 6 outlines utility and risk measures.

## 2 Functionality

NISS WebSwap is a Web service [1] that swaps one or more attributes (fields) between user-specified records in a microdata file, uploading the original data file from the user’s computer and downloading the file containing the swapped records.

The Web Services Description Language (WSDL) documentation [10] for NISS WebSwap is reproduced in Appendix A.

**Primary Functionality.** The principal functionality of NISS WebSwap is simultaneously to swap prescribed attributes in a specified fraction of the records in a database. Thus, the user specifies the:

**Data file** containing the microdata records. Details of the file format are discussed in §5.1;

**Swapping rate  $r$**  – is the fraction of records for which the specified attributes are initially marked to be swapped (see §3). Typically, that fraction is on the order of 1–10%; the rate in Table 1 is unrealistically high. At most  $2r\%$  of the records will actually be swapped; fewer will be swapped if records marked to be swapped are swapped with one another.

Details of how pairs of records to be swapped are selected appear in §3.

**Secondary Functionality.** NISS WebSwap provides additional capabilities to allow more refined implementation of data swapping, in the form of *constraints on unswapped attributes*, allowing the user to specify that two records will be swapped only if designated unswapped attributes are either:

**Equal:** for example, swapping may be allowed only for record pairs both drawn from the same state; or

**Different:** for example, records may be swapped only if they come from different counties. (If the data were to be aggregated to the county level for release, swapping within the same county would have no protective effect.)

Constraints are optional, and may be used in any combination. For the purposes of constraints, all attributes are treated as categorical.

## 3 The NISS WebSwap Algorithm

The NISS WebSwap swapping algorithm operates in the following manner.<sup>2</sup>

1. Initially, mark all records as unswapped, and set  $RS$ , the number of swapped records, to zero.
2. Randomly select a user-specified fraction (the `swap.rate` specification, see § 5.1) of the records and mark them as records to be swapped. Let  $RTS$  be the number of records to be swapped.
3. Select a record  $R_1$  at random from the current set of marked, unswapped records.
4. Select a second record  $R_2$  at random from the current set of *all* unswapped records (marked or not).
5. Determine whether the swap is a *true* swap. If not, return to Step 4.

---

<sup>2</sup>This is a general overview; fine details of the implementation are omitted.

6. Determine whether equality and inequality swapping constraints are satisfied. If not, return to Step 4. (If there does not exist any record  $R_2$  that is a feasible swap for  $R_1$ , it is recorded in the log file.)
7. Interchange the swapped attribute(s) between  $R_1$  and  $R_2$ , mark *both* as swapped, and set

$$RS = \begin{cases} RS + 1 & \text{if only } R_1 \text{ had been marked for swapping} \\ RS + 2 & \text{if both } R_1 \text{ and } R_2 \text{ had been marked for swapping.} \end{cases}$$

(If  $R_2$  had also been marked for swapping, then no record replaces it, which is why fewer than  $2 \times \text{swap.rate}\%$  of the records may actually be swapped.)

8. If  $RS < RTS$ , return to Step 3.

The algorithm is show pictorially in Figure 1.

## 4 Installation

### 4.1 System Requirements

Running the NISS WebSwap program requires that the Java 2 Platform, Standard Edition (J2EE) SDK 1.3.1 or 1.4 be installed on the user's computer. This can be downloaded from <http://java.sun.com> or from the Web site of operating system vendors. The NISS WebSwap program is a Java command-line application; therefore `java` must be in the system "path." An Internet connection is required to run the application, which uploads unswapped data to [www.niss-webservices.org](http://www.niss-webservices.org) and downloads swapped data to the user's computer.

### 4.2 Software Installation

1. Download NISS WebSwap distribution `NISSWebSwap.zip` from the NISS Web site, at [www.niss.org/WebServices/dg/WebSwap.html](http://www.niss.org/WebServices/dg/WebSwap.html).
2. Unzip distribution into separate directory, for example, named *NISSWebServices*.
3. Read the `README.webswap` file that is bundled with the distribution.

### 4.3 Demonstration Application

The NISS WebSwap distribution contains a demonstration application consisting of

- Data file `webswapdemo.orig`, an 8–attribute, 1024–record data set derived from the CPS, an excerpt of which is shown in Figure 3. See §2 and Table 1 for details such as attribute names.
- Specifications file `webswapdemo.specs`, which is shown in Figure 4. "Age" is the swapped attribute, and there are no constraints.

To run the demonstration, execute the script `webswapdemo.bat` (Windows) or `webswapdemo.sh` that is installed when the NISS WebSwap distribution is unzipped. This script contains the single command

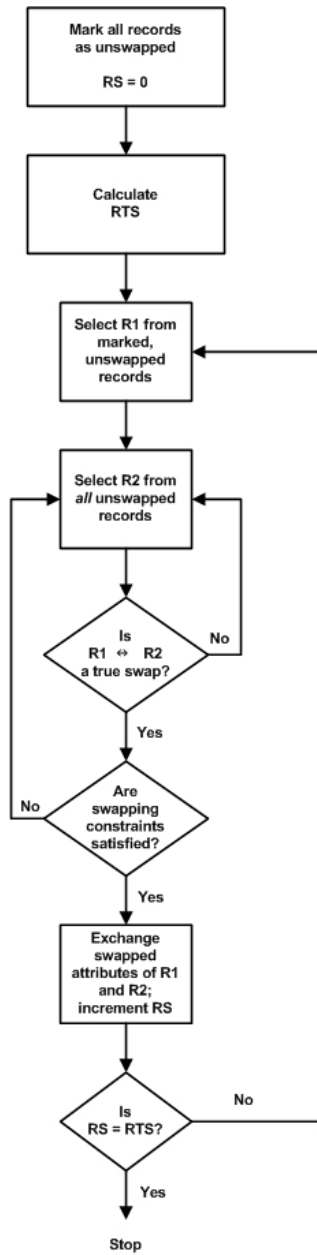


Figure 1: Flow chart of the NISS WebSwap algorithm.

```
[karr(12)E:/.../WebServices]> ./webswap.sh webswapdemo.specs
Checking specifications from file: webswapdemo.specs
Specifications OK.
Sending request to http://www.niss-webservices.org:8080/SwapServlet
Reading specs from webswapdemo.specs
Sending request to server....
Received response from server....
Writing files....
```

Figure 2: Screen shot of NISS WebSwap demonstration.

```
java -jar WebSwap-client.jar http://www.niss-webservices.org:8080/SwapServlet webswapdemo.specs;
```

see also §5.2. Successful execution of NISS WebSwap will produce the messages shown in Figure 2, an output file `webswapdemo.swapped` containing the swapped microdata, and a log file `webswapdemo.log`.

## 5 Usage

Once installed, NISS WebSwap is straightforward to run, in either demonstration mode (§4.3) or on the user's own files.

### 5.1 Input Files

Two input files are required by NISS WebSwap: a *data file* containing the microdata records and a *specifications file* specifying various parameters described below. Samples of both are contained in the NISS WebSwap distribution.

**Data File.** The file containing the microdata must be in comma-separated variable (CSV) format, with one record per line, as shown in Figure 3. CSV files can be produced by most database management systems and statistical packages (such as SAS), as well as Microsoft Excel.

*The first entry in each record must be a unique identifier*, even if one of the “real” attributes is a primary key to the database. The simplest way to create identifiers, as in Table 1 or (effectively) in Figure 3, is to number the records.

**Specifications File.** The specifications file is an ASCII text file containing the following items, in an *'item'='value'* format, one per line:

**num.records**, the number of records in the data file;

**data.file**, the name of the data file;

**log.file**, the name of the log file (see §5.3);

**output.file**, the name of the output file (see §5.3);

**swap.rate**, the swapping rate (see §2);

**attribute.specs**, a comma-separated list containing one entry for each attribute *other than the record identifier* in the data file. Allowable entries are:

```

1,25_55,Gov,Bach,UM,W,M,40,<50
2,25_55,SE,Bach,M,W,M,<40,<50
3,25_55,Pvt,HS,UM,W,M,40,<50
4,25_55,Pvt,<HS,M,NW,M,40,<50
5,25_55,Pvt,Bach,M,NW,F,40,<50
6,25_55,Pvt,Bach+,M,W,F,40,<50
7,25_55,Pvt,<HS,M,NW,F,<40,<50
8,25_55,SE,HS,M,W,M,40+,50+
9,25_55,Pvt,Bach+,UM,W,F,40+,50+
10,25_55,Pvt,Bach,M,W,M,40,50+

```

Figure 3: Sample CSV data file `webswapdemo.orig` included in the NISS WebSwap distribution. The file contains an 8-attribute data set derived from the CPS.

- S, indicating that the attribute is to be swapped;
- F, indicating that the attribute is to remain fixed — an equality constraint;
- D, indicating that the attribute must differ between any pair of records that are swapped — an inequality constraint;
- O, indicating other — the attribute is neither swapped nor is any constraint imposed on it.

There must be exactly one entry per attribute; this is the way in which NISS WebSwap determines the number of attributes.

**csv.type**, the type of the CSV data file. Allowable values are MS (the default) for Microsoft-CSV format (the option for data files saved from Excel spreadsheets), and ISO for ISO standard CSV format.

A sample specifications file `webswapdemo.specs`, which is contained in the NISS WebSwap distribution, is shown in Figure 4. The entries must be in the form shown there (e.g., `num.records = 1024`), and may appear in any order.

## 5.2 Running NISS WebSwap

**Command Line Execution.** To invoke NISS WebSwap from a command window, type

```
java -jar WebSwap-client.jar http://www.niss-webservices.org:8080/SwapServlet specs_file
```

at the command prompt, where *specs\_file* is the name of the specifications file (`webswapdemo.specs` for the demonstration).

**Script Execution.** The command scripts `webswap.bat` for Windows and `webswap.sh` for Linux/Unix contain the command

```
java -jar WebSwap-client.jar http://www.niss-webservices.org:8080/SwapServlet %1.
```

(This is the Windows version.) To execute NISS WebSwap with specifications file *specs\_file*, type

```
webswap specs_file.
```

```

num.records=1024
data.file=webswapdemo.orig
log.file=webswapdemo.log
output.file=webswapdemo.swapped
swap.rate=0.25
attribute.specs=S,0,0,0,0,0,0,0
csv.type=MS

```

Figure 4: Sample NISS WebSwap specifications file `webswapdemo.specs` included with the NISS WebSwap distribution. This file specifies that the first attribute — “Age” — is to be swapped. No other constraints are imposed.

```

1,25_55,Gov,Bach,UM,W,M,40,<50
2,25_55,SE,Bach,M,W,M,<40,<50
3,25_55,Pvt,HS,UM,W,M,40,<50
4,25_55,Pvt,<HS,M,NW,M,40,<50
5,55+,Pvt,Bach,M,NW,F,40,<50
6,25_55,Pvt,Bach+,M,W,F,40,<50
7,<25,Pvt,<HS,M,NW,F,<40,<50
8,25_55,SE,HS,M,W,M,40+,50+
9,<25,Pvt,Bach+,UM,W,F,40+,50+
10,25_55,Pvt,Bach,M,W,M,40,50+

```

Figure 5: Sample CSV NISS WebSwap output file arising from the data file in Figure 3 and specifications file in Figure 4. Comparison with Figure 3 shows that “Age” has been swapped for records 5, 7, and 9.

### 5.3 Output and Interpretation

NISS WebSwap produces and downloads to the user’s computer two files: an *output file* containing the swapped microdata and a *log file* containing details of the process. The names of these files are specified in the specifications file; for the demonstration, they are `webswapdemo.swapped` and `webswapdemo.log`.

**Output File.** The output file produced and downloaded by NISS WebSwap is in the same CSV format as the data file. An example is shown in Figure 5. Record identifiers are retained in the output file.

**Log File.** NISS WebSwap produces a log file each time it is run, which is illustrated in Figure 6.

### 5.4 Caveats

NISS WebSwap is a research prototype. While it has been tested extensively, it is not production software.

**Security.** File uploads and downloads by NISS WebSwap are not encrypted, and should be considered



```
Swapping Log: Mon Jun 10 11:23:19 EDT 2002
Number of risky records = 1024
Number of records marked for swapping = 256
Number of swaps performed: 218
```

Figure 6: Sample NISS WebSwap log file corresponding to the demonstration data file and specifications file appearing in Figures 3 and 4. Because only 218 swaps were performed, in 38 of these, both records swapped must have been marked for swapping. In the other 180 swaps, only one of the records had been marked.

insecure. Neither is the server `www.niss-webservices.org` absolutely secure. Therefore, NISS WebSwap should not be used on truly confidential data.

**Failure to Execute.** NISS WebSwap will fail to execute, possibly without informative error messages, if:

1. The server is down. This can be determined by pointing a Web browser to the URL `http://www.niss-webservices.org:8080/WebSwap/SwapIF`; if an error message is returned, then the server is down.
2. The specifications file or data file is missing from the directory from which NISS WebSwap is invoked (error messages will be produced in these cases).
3. The data file is not well-formed CSV (for example, contains entries with internal commas that are not protected by quotation marks).
4. The total size of uploaded data exceeds the maximum allowable upload size. The allowable upload size is subject to change depending on the security configuration on the server; currently data files with data size (the number of records times the number of attributes) less than 50,000 and file size less than 2MB are acceptable.
5. Any record in the data file contains more or fewer than the number of attributes specified by the `attribute.specs` entry in the specifications file.

It may also fail under other circumstances.

## 6 Examples of Utility and Risk

As mentioned in Section 1, a risk-utility framework can be used to evaluate the result of any SDL procedure. This section presents an example of utility measure, and one of a risk measure, appropriate for data-swapping.

In Section 1, we mentioned that data swapping may change the joint probability distribution of sets of attributes containing the swapped attributes. It is thus natural to use a measure of the change in joint distribution as a measure of (dis)utility or data distortion created by the swapping process. The Hellinger distance (see [7] for a definition) is one such measure of “distance” between two (possibly multivariate) distributions. In particular, for any specified collection of  $k$  attributes in the data, the  $k$ -way Hellinger

distance measures distance between pre- and post-swap distributions of a specified subset of  $k$  attributes in the data. In our research we have used this  $k$ -way distance to systematically study the effect of data swapping on different sets of attributes.

A widely held viewpoint in the SDL area is that records with rare attribute values are susceptible to identity disclosure. Well-established rules such as the so-called  $n$ -rule<sup>3</sup> embody this viewpoint and are often used to ascertain the risk of a data release. Based on this line of reasoning, we can devise a measure of the risk of release by constructing a cross-tabulation of the post-swap data and using the number of *unswapped records* that fall in cells with low counts as a measure of risk.

Although neither of the above computations is part of NISS WebSwap itself, the Web site at [www.niss.org/WebServices/dg/WebSwap.html](http://www.niss.org/WebServices/dg/WebSwap.html) will contain links to programs for computing  $k$ -way Hellinger distances, and risk of release.

## 7 Support

Limited support for NISS WebSwap is available by sending E-mail to [webswap@niss.org](mailto:webswap@niss.org). Please use NISS WebSwap as the subject line.

## 8 Acknowledgements

The authors would like to thank Adrian Dobra for his comments. Support for the research was provided by National Science Foundation grant EIA-9876619 to NISS and by the National Center for Education Statistics (NCES).

## References

- [1] E. Cerami. *Web Services Essentials*. O'Reilly & Associates, Sebastopol, CA, 2002.
- [2] T. Dalenius and S. P. Reiss. Data swapping: A technique for disclosure control. *J. Statist. Planning Inf.*, 6:73-85, 1982.
- [3] G. T. Duncan, T. B. Jabine, and V. A. de Wolf, editors. *Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics*. National Academy Press, Washington, 1993. Report of a Panel on Confidentiality and Data Access, Committee on National Statistics.
- [4] G. T. Duncan, S. A. Keller-McNulty, and S. L. Stokes. Disclosure risk vs. data utility: The R-U confidentiality map. *Management Sci.*, 2001. Under review.
- [5] Federal Committee on Statistical Methodology. *Report on Statistical Disclosure Limitation Methodology*. US Office of Management and Budget, Washington, 1994.
- [6] S. Hettich and S. D. Bay. The UCI KDD Archive. Irvine, CA: University of California, Department of Information and Computer Science, 1999. Available on-line at [kdd.ics.uci.edu](http://kdd.ics.uci.edu).

---

<sup>3</sup>The  $n$ -rule stipulates that, in a tabulation of the data, records that fall in any cell with a count less than or equal to  $n$  should be considered to be at risk. The threshold is usually taken to be a small number such as  $n = 2$ .

- [7] L. Le Cam and G. L. Yang. *Asymptotics in Statistics*. Springer–Verlag, New York, 1990.
- [8] L. C. R. J. Willenborg and T. de Waal. *Statistical Disclosure Control in Practice*. Springer–Verlag, New York, 1996.
- [9] L. C. R. J. Willenborg and T. de Waal. *Elements of Statistical Disclosure Control*. Springer–Verlag, New York, 2001.
- [10] World Wide Web Consortium. Web services description language (wsdl), 2002. Information available on line at [www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl).

## A The NISS WebSwap WSDL

```
<?xml version="1.0" encoding="UTF-8"?>

<definitions name="Swap_dataService"
targetNamespace="http://WebSwap_swap.org/wsdl"
xmlns:tns="http://WebSwap_swap.org/wsdl"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:ns2="http://WebSwap_swap.org/types"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <types>
    <schema targetNamespace="http://WebSwap_swap.org/types"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:tns="http://WebSwap_swap.org/types"
      xmlns:soap-enc="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="SwapData">
        <sequence>
          <element name="numFields" type="int"/>
          <element name="outputFile" type="string"/>
          <element name="numRecords" type="int"/>
          <element name="riskCutoff" type="double"/>
          <element name="data" type="tns:ArrayOfArrayOfstring"/>
          <element name="dataFile" type="string"/>
          <element name="constraints" type="base64Binary"/>
          <element name="log" type="tns:ArrayOfstring"/>
          <element name="swapRate" type="double"/>
          <element name="riskFraction" type="double"/>
          <element name="logFile" type="string"/>
          <element name="csvType" type="string"/></sequence></complexType>
      <complexType name="ArrayOfArrayOfstring">
        <complexContent>
          <restriction base="soap-enc:Array">
            <attribute ref="soap-enc:arrayType"
              wsdl:arrayType="tns:ArrayOfstring[]"/>
          </restriction></complexContent></complexType>
      <complexType name="ArrayOfstring">
        <complexContent>
          <restriction base="soap-enc:Array">
```

```

        <attribute ref="soap-enc:arrayType"
            wsdl:arrayType="string[]" /></restriction>
        </complexContent></complexType></schema></types>
<message name="doSwap">
    <part name="SwapData_1" type="ns2:SwapData" /></message>
<message name="doSwapResponse">
    <part name="result" type="ns2:SwapData" /></message>
<portType name="SwapIF">
    <operation name="doSwap">
        <input message="tns:doSwap" />
        <output message="tns:doSwapResponse" /></operation></portType>
<binding name="SwapIFBinding" type="tns:SwapIF">
    <operation name="doSwap">
        <input>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                use="encoded" namespace="http://WebSwap_swap.org/wsdl" /></input>
        <output>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                use="encoded" namespace="http://WebSwap_swap.org/wsdl" /></output>
        <soap:operation soapAction="" /></operation>
    </binding>
<service name="Swap_data">
    <port name="SwapIFPort" binding="tns:SwapIFBinding">
        <soap:address location="http://www.niss.web-services:8080/WebSwap/SwapIF" />
    </port></service></definitions>

```